UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

ARCHITECTING A STRATEGIC ENGINEERING FRAMEWORK FOR EVOLVING CYBER-PHYSICAL-SOCIAL SYSTEMS

A DISSERTATION
SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the
Degree of

DOCTOR OF PHILOSOPHY

By MAYANK JAYANT BHALERAO
Norman, Oklahoma
2026

ARCHITECTING A STRATEGIC ENGINEERING FRAMEWORK FOR EVOLVING CYBER-PHYSICAL-SOCIAL SYSTEMS

A DISSERTATION APPROVED FOR THE
SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Janet K. Allen, Co-Chair

Dr. Farrokh Mistree, Co-Chair

Dr. David Ebert

Dr. John Klier

Dr. Charles Nicholson

©Copyright by MAYANK JAYANT BHALERAO 2026
All Rights Reserved.

Acknowledgments

I am truly grateful for the support I received while completing this dissertation, and I deeply appreciate every individual who contributed to its realization.

I am sincerely thankful to my advisors, Professor Janet K. Allen and Professor Farrokh Mistree, for their unwavering support and exceptional mentorship throughout this period. They guided me not only in shaping the intellectual contributions of this dissertation and in ensuring its timely completion, but also in fostering growth well beyond the realm of research, helping me develop new competencies and embody values, ethics, and discipline through structured and organized effort. Through their wisdom and intellectual prowess they enlightened my path and challenged me to reach new heights in my personal and professional life. I extend my deepest gratitude to them for giving me the opportunity to establish my own identity as a researcher, and for allowing me to nurture its distinctiveness with such dedication.

I am thankful to my committee members Dr. David Ebert, Dr. John Klier, and Dr. Charles Nicholson. I am thankful to my committee members. I am grateful to Dr. David Ebert for his constructive feedback and his suggestions on the interpretability of systems and the purposeful integration of intelligence, through which he sharpened the way I frame what a designer should be able to see and act upon. I am thankful to Dr. John Klier for his advice on research directions, for our discussions on AI models and the impact of my research, and for his counsel on the future course of my career. I am especially thankful to Dr. Charles Nicholson for his feedback on setting the research layout and for his guidance on other professional activities.

A big thank you to everyone at the Systems Realization Laboratory @ OU, colleagues who became friends along the way. In our weekly meetings and in our discussions across the whiteboard they strengthened this work, they read drafts on short notice, and they kept me good company through the long years. I am fortunate to have grown alongside such a group.

I am thankful to the School of Industrial and Systems Engineering at the University of Oklahoma for its support throughout my studies, and to its administrative staff in particular, who attended to countless matters with care and efficiency and let me keep my attention where it belonged.

Above all, I owe a profound debt of gratitude to my parents, who have shaped me into the person I am today. Whatever I have achieved rests on foundations they laid long before I understood they were laying them, through their sacrifices, their quiet persistence, and the countless prayers they offered on my behalf. They gave me the ground beneath every success I claim, and those successes are, in truth, as much theirs as mine. I am also deeply grateful to my sister and brother-in-law for their steadfast support. They allowed me to chase my dreams without hesitation, encouraged me through every obstacle, and cheered me toward every achievement.

I am also thankful to all my friends, near and far, for their support through these years. I am grateful for their patience, and for the faith they kept in me.

I am profoundly grateful for the generous financial support provided by the John and Mary Moore Chair and the L.A. Comp Chair at the University of Oklahoma. With this support I was able to complete the work presented in this dissertation.

Deepest gratitude to the Divine, in whose grace every step of this path was taken.

TABLE OF CONTENTS

ACKNOWLEDGMENTS iv

TABLE OF CONTENTS vi

LIST OF TABLES xix

LIST OF FIGURES xxii

ABSTRACT xxvi

CHAPTER 1: Motivation and Problem Identification 1

1.1 Introduction to the Challenge of Designing Evolving Cyber- Physical-Social Systems 1

1.2 Strategic Engineering as the Paradigmatic Frame 5

1.3 Two Fundamental Asymmetries in E-CPSS Design 7

1.3.1 Information Asymmetry 7

1.3.2 Intelligence Asymmetry 8

1.4 Six Escalating Challenges in E-CPSS Design 9

1.5 The Decision-State Information Model as the Synthesis Target 17

1.6 Research Questions and Hypotheses 19

1.7 Two Perspectives on the Dissertation's Progression 25

1.7.1 Capability Escalation 25

1.7.2 Knowledge Escalation 26

1.8 Goals and Focus in Dissertation 27

1.8.1 Problem to be Addressed 27

1.8.2 Contributions Through This Work 28

1.8.3 Way Forward 31

1.9 Validation Strategy and the V&V Square 31

1.10 Organization of the Dissertation 33

1.11 What We Have Presented in Chapter 1 and What Comes Next 35

References 35

CHAPTER 2: Critical Literature Review and Justification of Research Questions.....38

Background and Context.....38

Preface.....39

Glossary.....40

2.1 Frame of Reference.....41

2.1.1 Integrating Human Factors into the CPSS.....42

2.1.2 Addressing the Entities: Cyber, Physical, and Social Spaces.....42

2.1.3 E-CPSS Foundations: Objectives and Organization.....44

2.1.4 Paradigm-Level Evolution by Means of the Knowledge Spaces.....46

2.1.5 Indicators and Metrics for Adaptation and Evolution.....49

2.1.6 Micro, Meso, and Macro Scales of Adaptation.....51

2.1.7 The Amplifying Role of Social Space.....52

2.1.8 Method of the Study.....52

2.1.9 Towards an Implication for Design.....55

2.1.10 Verification and Validation.....56

2.2 Design and Decision Making in the E-CPSS.....60

2.2.1 Importance of the Social Space.....60

2.2.2 Social Space Literature Highlights.....62

2.2.3 Evolution in Social Space.....67

2.3 Evolving Cyber-Physical-Social System.....68

2.3.1 Summary of State of the Art and the Foundational Insights for an E- CPSS.....68

2.3.2 Characteristics and Functional Requirements.....79

2.3.3 Behavioral and Structural Requirements.....88

2.4 E-CPSS Cognitive Map for Decision-Making.....93

2.4.1 Features of the Cognitive Map.....96

2.4.2 Workings of the Cognitive Map.....98

2.5 Key Findings in Designing E-CPSS.....100

2.6 Closing Remarks for Chapter 2.....102

2.6.1 Scholarly Propositions: Synthesizing Key Contributions....104
2.6.2 Limitations and Deficiencies....105
2.6.3 Near-Future Research Opportunities....106
2.6.4 Final Reflections: Toward a Cognitively Enriched Design Epistemology....109
References....110
Appendix for Chapter 2....123

CHAPTER 3: Observing E-CPSS: CPSS-Structured Predictive Modeling for Sociotechnical System Design....126
Background and Context....126
Preface....127
Research Question, Hypothesis and New Knowledge in This Chapter....128
3.1 Frame of Reference....129
3.1.1 Overview of Bird Strikes and Their Significance....130
3.1.2 Discussion of Artificial Light at Night (ALAN) as a Contributing Factor....131
3.1.3 Application to Public Policy Design and Mitigation Strategies....131
3.1.4 Organization of the Chapter....132
3.2 Review of Related Work....133
3.2.1 Predictive Modeling for Geospatial Data....134
3.2.2 CPSS as a Paradigm for Modeling Emergent Behavior....135
3.2.3 Research Gap Identification....136
3.3 Method: Impact of ALAN on Bird Strikes....136
3.3.1 Description of Data Sources....139
3.3.2 Methods for Preparing and Integrating Data....143
3.3.3 Exploratory Data Analysis (EDA) Techniques....143
3.3.4 Statistical and Machine Learning Techniques for Predictive Modeling....144
3.4 Results....146
3.4.1 Exploratory Data Analysis (Eda) Findings....146
3.4.2 Correlation Analysis of Bird Strikes and Light Pollution....150

3.4.3 Predictive Modeling Outcomes....151
3.5 Discussion....154
3.5.1 Results Interpretation....154
3.5.2 Limitations....156
3.5.3 Generalizability of the Modeling Approach....157
3.6 Closing Remarks for Chapter 3....160
References....162

CHAPTER 4: Designing Robust Policies for Evolving Cyber-Physical-Social Systems in Uncertain Environments....168
Background and Context....168
Preface....169
Research Question, Hypothesis, and New Knowledge in This Chapter....170
4.1 Frame of Reference....171
4.2 Foundations for Effective Policy Designing....173
4.2.1 Uncertainty, Robustness, and Policy Designing....175
4.2.1.1 Uncertainties in Policy Designing....176
4.2.1.2 Understanding Robustness in the Context of Policy Designing and Uncertainty....179
4.2.1.3 Designing Policy Robustness for Uncertain Environments....180
4.2.2 Towards Multi Level Decision Environment for Policy Designing....181
4.3 Test Problem for Validation: Agriculture 5.0....183
4.3.1 Significance of Agriculture 5.0 for Policy Designing....183
4.3.2 Importance of the Selected Test Problem with Respect to Policy Designing....184
4.3.3 Assumptions....185
4.4 Proposed Framework and Method....185
4.4.1 Detailed Description of the Framework....186
4.4.2 Understanding the Roots, Concepts, and Ideation....188

4.4.2.1 Considerations for Model Based Policy Designing and Robustness and Uncertainties 190
4.4.2.2 Balancing Abstraction and Action in Policy Designing....192
4.4.3 The Coupled Decision Method 194
4.5 Uncertainty Management and Incorporating Robustness in Policy Designing 197
4.5.1 Design Capability Index 198
4.6 Demonstration of the Efficacy of the Proposed Framework....200
4.6.1 Mathematical Formulation....202
4.6.2 Demonstration of Results....207
4.6.3 Interpretation and Discussion on Results....209
4.6.4 Sample Policy Design 210
4.7 Closing Remarks for Chapter 4....212
References....214

CHAPTER 5: Designing Adaptive Strategies: Goal- Oriented Inverse Design across E-CPSS State Transitions....225
Background and Context....225
Preface....226
Research Question, Hypothesis, and New Knowledge in This Chapter....227
5.1 Frame of Reference....228
5.1.1 Designing Policies as Evolving Cyber Physical Social Systems....229
5.1.2 Towards Robustness in Designing Policy....231
5.1.3 Understanding the Temporal and Related Aspects 233
5.2 Understanding the Foundations of the Proposed Method....236
5.2.1 The Compromise Decision Support Problem 236
5.2.2 Modelling Uncertainties and Evaluation Metrics 237
5.2.3 Interpretable Self-Organizing Maps (iSOM) 239
5.3 Goal-Oriented Inverse Design Method (GOIDM) for Robust Policy Designing....240
5.4 Validation Through the Test Problem 246

5.4.1 Problem Statement Data Source 247
5.4.2 Establish Forward Modelling of the Goal-Oriented, Inverse Method 248
5.4.2.1 Defining Current, Final, and Transitional States of the System 248
5.4.2.2 Defining Historical Trends....249
5.4.2.3 Projecting Future Goals and Designing Transitional States 250
5.4.3 Formulating the Design Space Using Inverse Design Principles 251
5.4.3.1 Mathematical Formulation for the Final State (Year 2030)....251
5.4.3.2 Computational Results and Robust Design Exploration for the Final State (Year 2030) 256
5.4.3.3 Mathematical Formulation for the Transition State (Year 2023)....259
5.4.3.4 Computational Results and Robust Design Exploration for the Transition State (Year 2023) 260
5.4.4 Adaptive Policy Updates Through Feedback Mechanisms 261
5.4.5 Iterative Refinement and Final Transition to the Target State 266
5.4.6 Synthesizing the Approach: GOIDM as E-CPSS 268
5.5 Closing Remarks for Chapter 5....272
References....275

CHAPTER 6: Augmenting Design Intelligence: Generative AI and Knowledge Graphs for Interpretable Trade-Off Navigation....281
Background and Context....281
Preface....282
Research Question, Hypothesis, and New Knowledge in This Chapter....283
6.1 Frame of Reference....284
6.1.1 Background in Multi-Objective Design and Satisficing Decisions....284
6.1.2 Gaps in Scalability, Interpretability, and Adaptivity of Current Approaches....285
6.1.3 Opportunities at the Intersection of Generative AI and Design Decision-Making 288
6.1.4 Positioning of the Proposed Work 291
6.2 Proposed Framework 292

6.3 Test Problem: Hot Rod Rolling (HRR) 296
6.4 Results and Discussions 299
6.4.1 Knowledge Extraction from Data (Step 1) 299
6.4.2 Knowledge Exploration (Step 2) 309
6.4.2.1 Interactive Querying (Step 2.1) 309
6.4.2.2 Scenario Clustering and Narratives (Step 2.2) 310
6.4.3 Adaptable Solution-Space Generation (Step 3) 311
6.4.4 Results and Discussion 317
6.5 Closing Remarks for Chapter 6 318
References 321

CHAPTER 7: Autonomous Strategy Execution: Knowledge-Governed Agents for Adaptive Decision-Making in E-CPSS 325

Background and Context 325
Preface 327
Research Question, Hypothesis, and New Knowledge in This Chapter 329
7.1 Frame of Reference 329
7.2 Related Work 333
7.2.1 Satisficing and Risk-Sensitive Reinforcement Learning 334
7.2.2 Where the Aspiration Comes From 335
7.2.3 Knowledge Graphs and Structured Information in Reinforcement Learning 337
7.2.4 Reward Shaping and the Distinction from Compiled Aspiration 338
7.2.5 Interpretable and Auditable Reinforcement Learning 339
7.2.6 Positioning Summary: What We Borrow and What is New 340
7.3 Problem Setting 342
7.3.1 Markov Decision Processes 342
7.3.2 The Inventory Management Running Example 343
7.3.3 Knowledge Graph and Facts 344
7.3.4 The Compiler and the Aspiration 345

7.3.5 Putting the Pieces Together 346
7.4 The Compiled-Aspiration MDP 347
7.4.1 The Compiled-Aspiration MDP 347
7.4.2 Axioms of an Admissible Compiler 348
7.4.3 The Graph-Structured Extension 349
7.4.4 The Concrete Compiler 351
7.4.5 A Worked Example 352
7.5 The CA-RS-Q Algorithm 353
7.5.1 Quantities Maintained During Execution 356
7.5.2 Computing the Risk-Sensitive Value 357
7.5.3 Selecting an Action 358
7.5.4 Updating the Value Estimate and the Reliability Count 358
7.5.5 The Role of the Aspiration 359
7.5.6 Behavior under an Evolving Knowledge Graph 360
7.5.7 Per-Step Cost 360
7.5.8 Relationship to Prior Algorithms 360
7.6 Theoretical Results 361
7.6.1 Theorem 1: Convergence under an Eventually Stable Knowledge Graph 361
7.6.2 Theorem 2: Finite-Time Satisficing Rate 364
7.6.3 Theorem 3: Robustness under Flat Knowledge Graph Perturbation 366
7.6.4 Theorem 3-G: Robustness under Graph-Structured Perturbation 368
7.7 Results and Empirical Evaluation 372
7.7.1 Benchmarks 373
7.7.2 Verification of Theorem 1: Convergence under an Eventually Stable Knowledge Graph 373
7.7.3 Verification of Theorem 2: Satisficing Rate under a Stationary Knowledge Graph 376
7.7.4 Verification of Theorem 3: Robustness under Knowledge-Graph Perturbation 379
7.7.5 Catastrophe Protection under Structural Shock 380

7.7.6 Cross-Benchmark Comparison 385
7.7.7 A Worked Decision Card: Auditability 386
7.7.8 Summary of Empirical Findings 390
7.8 Discussion 390
7.8.1 Positioning the Empirical Evaluation 390
7.8.2 What the Baseline Comparisons Reveal 391
7.8.3 Where the Framework's Structural Advantages Become Measurable 395
7.8.4 Threats to Validity 396
7.8.5 Future Work 397
7.9 Closing Remarks for Chapter 7 398
References 400

CHAPTER 8: Assessing Adaptive Interventions: Cognitively Grounded Stakeholder Simulation in E-CPSS 405
Background and Context 405
Preface 407
Research Question, Hypothesis, and New Knowledge in This Chapter 408
8.1 Frame of Reference 409
8.1.1 Simulating Stakeholder Deliberation for Strategy and Policy Decisions 409
8.1.2 Why Current Generative Simulations are Not Yet Decision-Grade 410
8.1.3 The Proposed Architecture and Contributions 411
8.2 Related Work 413
8.2.1 Cognitive Architectures for Language Agents 414
8.2.2 Emotion and Affect in Artificial Agents 414
8.2.3 Memory in LLM Agents 415
8.2.4 Neuro-Symbolic and Auditable Multi-Agent Systems 416
8.3 The Proposed Cognitive Architecture 417
8.3.1 Two Reasoning Layers and the Per-Turn Cycle 418
8.3.2 The World Model 420

8.3.3 Mood and Arousal 421
8.3.4 Affect-Modulated Memory 423
8.3.5 Mood Feedback and Recovery 424
8.3.6 Cognitive Biases as Four Addressable Layers 425
8.3.7 The Neuro-Symbolic Rule Engine 425
8.3.8 Orchestration and Social Influence 427
8.3.9 Stakeholder Personas 428
8.3.10 The Carbon-Policy Deliberation as a Test Problem 429
8.3.11 Experimental Design and Reproducibility 430
8.3.12 Evaluation Measures 431
8.4 Results and Discussion 432
8.4.1 Memory is the Most Important Mechanism for Coherent Behavior (C4) 433
8.4.2 Arousal Makes Agents Behave More Consistently (C1) 437
8.4.3 Each Stakeholder Class Departs from Rational Behavior by a Measured Amount (C3) 438
8.4.4 The Symbolic Layer Enforces Real, Auditable Constraints (C2) 441
8.4.5 Emotion Dynamics Follow Theory and Produce Stable Opposing Factions 444
8.4.6 The Findings Hold up under Stress Testing 449
8.4.7 The Architecture Transfers to a Stronger Model from a Different Family 450
8.4.8 The Same Architecture Produces Very Different Agents 452
8.4.9 Scope of the Reported Evidence 454
8.5 Discussion 455
8.5.1 Insights on Demonstration Through the Architecture 455
8.5.2 Computational Complexity and Cost 456
8.5.3 Limitations 458
8.5.4 Future Research Avenues and Open Research Questions 459
8.6 Closing Remarks for Chapter 8 460
Appendix 8A. Parameters, Biases, Roster, and Scenarios 461
References 467

CHAPTER 9: Synthesis and Closure 470

9.1 Objectives in This Chapter 470

9.2 A Summary of the Dissertation 470

9.3 The Decision-State Information Model, Formal Definition 481

9.3.1 S, the Set of States 482

9.3.2 I(si), the Structured Tri-Space Characterization 482

9.3.3 D(si) and D(si to Si+1), the Decision Maps 483

9.3.4 M(si), the Method Set 484

9.3.5 K(si), the Knowledge We Produce and Carry Forward 485

9.3.6 T(si to Si+1), the Transition Condition 485

9.4 Why These Five Layers 489

9.4.1 The Sense in which the Framework is Sufficient 489

9.4.2 The Two Asymmetries, Closed Layer by Layer 490

9.5 The Functionalities We Add 493

9.5.1 The Functionality of Each Method 494

9.5.2 The Functionalities We Add Through Composition 495

9.5.3 What the Functionalities Amount To 497

9.6 Operationalizing the Eleven E-CPSS Design Requirements 500

9.6.1 Functional Requirements 500

9.6.2 Behavioral Requirements 502

9.6.3 Structural Requirements 503

9.6.4 The Completeness of the Mapping 504

9.7 Answering the Research Questions and Evaluating the Hypotheses 509

9.7.1 Observation, RQ1 510

9.7.2 Decision Structuring, RQ2a 511

9.7.3 Adaptive Design, RQ2b 512

9.7.4 Augmented Intelligence, RQ3a 513

9.7.5 Autonomous Execution, RQ3b 514
9.7.6 Social Assessment, RQ4 516
9.7.7 The Evidence across the Six Questions 517
9.8 Theoretical Performance Validity 520
9.8.1 The Test Problems as Representatives of General Classes 520
9.8.2 The Completeness of the Framework across Unrelated Domains 522
9.8.3 The Invariant Structure of Each Method 522
9.8.4 The Scope and the Limit of the Generalization Argument 524
9.9 Answering the Principal Research Question 527
9.9.1 The Scope of the Answer 528
9.10 Contributions and New Knowledge 529
9.10.1 The Method-Level Contributions 531
9.10.2 The Synthesis Contribution 534
9.10.3 What a Designer Can Do Through the Framework 534
9.11 Limitations and Opportunities for Improvement 538
9.11.1 The Scope Limitation 538
9.11.2 Method-Level Limitations 538
9.11.3 Framework-Level Limitations 541
9.12 A Vision for Future Research 542
9.12.1 Consolidating and Completing the Framework 542
9.12.2 Research Avenues We Open Through the Dissertation 543
9.12.3 Toward Self-Designing Evolving Systems 545
9.13 Perspectives on the Progression Revisited 548
9.13.1 Capability Escalation 549
9.13.2 Knowledge Escalation 549
9.14 I Statement 551
9.14.1 Self-Reflection 551
9.14.1.1 Where I Began 551

9.14.1.2 The Question I Keep Returning To 552
9.14.1.3 How I Came to Work on Artificial Intelligence 553
9.14.1.4 Four Ways My Thinking Developed 555
9.14.1.5 The Question I Carry Forward 556
9.14.1.6 Deciding with Incomplete Information 557
9.14.1.7 The Idea I Expect to Carry Furthest 558
9.14.1.8 How I Have Grown 558
9.14.1.9 Janet, Farrokh, and the Systems Realization Laboratory 560
9.14.2 Self-Assessment Against Research Competencies 560
9.14.2.1 Where I Intend to Grow Next 562
9.14.3 My Vision: Engineering Intelligence 564
9.14.3.1 The Question I Intend to Work 564
9.14.3.2 Three Positions I Hold 564
9.14.3.3 Thrust 1: Context as a First-Class Object in Reinforcement Learning and Decision Theory 565
9.14.3.4 Thrust 2: Governable World Models 566
9.14.3.5 Thrust 3: Cognitive Architectures for Machine Judgment 567
9.14.3.6 Why I Treat These as One Problem 567
9.14.3.7 On General Intelligence for Engineered Systems 568
9.14.3.8 How I Intend to Take This to the World 569
9.14.4 Closing 570
9.14.5 Publications Based on This Dissertation 570

LIST OF TABLES

Table 1.1: Research Questions to Chapters and Validation Quadrants 11

Table 2.1: Examples of Social Entities 66

Table 2.2: State of the Art and Foundational Insights on E-CPSS 71

Table 2.3: Characteristics of an E-CPSS 83

Table 3.1: Dual-Unit Representation of Light Pollution Data 141

Table 3.2: Training Dataset for Predictive Model 145

Table 3.3: List of Variables and Their Distribution 145

Table 3.4: Monthly Averages of Performance Metrics in the Training Dataset 152

Table 4.1: Key Literature Review: Comparing Policy Designing Frameworks 182

Table 4.2: System Variables and Its Significance for Agriculture 5.0 201

Table 4.3: Sample Scenarios for Weights 207

Table 4.4: Solution Points Selected in RoI as per Hits (DCI Converted Values for G1 and G2) 209

Table 4.5: Sample Policy Design 211

Table 5.1: Projected Values for the Goals in the Transitional State (Year 2023) 251

Table 5.2: System Variables, Ranges, and Variability (No. of Employees) 252

Table 5.3: Sample Scenarios with Weight Assignments and Corresponding Results for the Final State (the Year 2030) 257

Table 5.4: Sample Scenarios and Robust Solution Points for the Final State 258

Table 5.5: Adaptation Rate Calculation between the Years 2030 and 2023 259

Table 5.6: System Variables, Ranges, and Variability for State 2023 (No. of Employees) 260

Table 5.7: Relaxation Rates and Adjusted Goal Values for Feasibility 261

Table 5.8: Sample Scenarios and Robust Solution Points for Transition State 2023 262

Table 5.9: Targeted and Achieved System Goals for the Year 2023 262

Table 5.10: Updated System Goals and Weights for the Final State (Year 2030) 263

Table 5.11: Scenario Adjustments for Adaptive Policy Design 264

Table 5.12: Comparison of Employment Distribution by Sector in Scenarios 27 and 28 (No. of Employees) 266

Table 5.13: Updated Goals for Transitional State (Year 2027) 267

Table 5.14: Robust Solution (Number of Employees) for the Transition State Year 2027 268

Table 5.15: Comparison of GOIDM with Traditional Goal Programming–Based Policy Design Approaches 270

Table 6.1: Decision Variables of Design Level 1 (Materials Designer).....297

Table 6.2: Performance Goals of Design Level 1 (Materials Designer).....298

Table 6.3: Decision Variables of Design Level 2 (Cooling Process Designer).....298

Table 6.4: Performance Goals of Design Level 2 (Cooling Process Designer).....298

Table 6.5: Structured JSON Scenario Examples and Few-Shot Prompt Used for Knowledge Extraction.....300

Table 6.6: Sample of Quantitative Validation of Extracted Knowledge Statements 302

Table 6.7: Set of Knowledge Extracted by the LLM from the Curated HRR Scenario Dataset..303

Table 6.8: Example Fine-Tuning Data for Schema-Constrained Knowledge-Graph 306

Table 6.9: Knowledge Graph Relationship Types and Associated Attributes.....307

Table 6.10: LLM Input-Output Showing No Feasible Solution 313

Table 6.11: LLM Input-Output for Identifying Satisficing Solutions (Relaxed Thresholds) 314

Table 6.12: Common Satisficing Solutions Identified across Design Levels 1 and 2 315

Table 6.13: LLM Input-Output for Identifying Updated Satisficing Solutions 316

Table 6.14: Updated Satisficing Solutions Identified across Design Levels 1 and 2.....317

Table 7.1: Borrowed Versus New Components in the Present Chapter.....340

Table 7.2: Summary of Theorems.....371

Table 7.3: Q-Error Trajectory $\|Q - Q^*\|$ on Benchmark A (10 Seeds) 374

Table 7.4: Extended Q-Error Trajectory $\|Q - Q^*\|$ on Benchmark B (5,000 Episodes, 10 Seeds) 375

Table 7.5: Satisficing Rate and Empirical Gap G_emp (10 Seeds) 377

Table 7.6: (G+) Diagnostic on Benchmarks A and B.....377

Table 7.7: Robustness Verification across All Three Benchmarks (10 Seeds) 379

Table 7.8: Post-Shock Catastrophe Rate with 95 Percent Wilson Confidence Intervals (10 Seeds; 3,000 Post-Shock Episodes on A and B, 10,000 on C).....382

Table 7.9: Compiler Ablation (10 Seeds).....385

Table 7.10: Cross-Benchmark Scale Comparison 386

Table 7.11: Decision Card for the Spotlight Week (Week 39) of a Benchmark B Episode.....388

Table 7.12: Headline Measurements across Four Baseline-Comparison Scenarios (10 Seeds) ..391

Table 8.1: Capability Ladder under Design C, Reported as Means over the Four Scenarios (Source: Full 136-Run Grid).....434

Table 8.2: Mean Regret by Stakeholder Class for the Full Architecture under Design C 439

Table 8.3: Constraint Integrity: The Number of Rule Firings Actively Enforced per Run, with the Symbolic Layer Enabled Versus Disabled (Source: Full Grid) 442

Table 8.4: Robustness of the Hybrid-Scenario Results under Three Perturbations (Source: Robustness Study, Recomputed from Raw Run Records).....450

Table 8.5: Cross-Model Comparison under Matched Constrained Decoding 451

Table 9.1: Chapter-By-Chapter Summary of Design Challenges, Methods, Contributions, and New Knowledge.....477

Table 9.2: The Components of the Decision-State Information Model.....488

Table 9.3: Closure of the Two Asymmetries, Layer by Layer 492

Table 9.4: The Functionalities We Add, by Method and by Composition.....498

Table 9.5: Operationalization of the Eleven E-CPSS Design Requirements across the Five Layers 506

Table 9.6: Evaluation of the Hypotheses across the Verification and Validation Square 518

Table 9.7: Problem Class Characteristics and Invariant Method Structure 525

Table 9.8: Contributions and New Knowledge of the Dissertation 536

Table 9.9: Self-Assessment Against the Competencies I Developed 560

LIST OF FIGURES

Figure 1.1: Dissertation Progression 5
Figure 1.2: Decision State Information Model 19
Figure 1.3: Verification and Validation Square 32
Figure 1.4: Dissertation Roadmap 35
Figure 2.1: Cyber-Physical-Social Spaces and Interactions 44
Figure 2.2: Holistic Social Space of CPSS 64
Figure 2.3: Evolving Problem Structures and Decision Strategies 68
Figure 2.4: Distinction Among CPSS, CPHS, and SCPS 81
Figure 2.5: Classification of Design Requirements 86
Figure 2.6: Cognitive Map for the E-CPSS 96
Figure 3.1: Conceptual Map: Policy Design for the Impact of ALAN on Birds as a CPSS 137
Figure 3.2: Addressing Bird Strikes Through a Model-Based, Data-Driven Systems Design 139
Figure 3.3: Diagram of Predictive Modeling Approach 140
Figure 3.4: Yearly Bird Strikes Trend (1990-2023) 147
Figure 3.5: Recorded Bird Strikes by Time of Day (1990- 2023) 147
Figure 3.6: Species-Specific Analysis (Top 10 Species Involved in Bird Strikes) 148
Figure 3.7: Distribution of Bird Strike Occurrences by Day of Year (1990- 2023) 149
Figure 3.8: Recorded Bird Strike Locations in the United States (1990-2023) 149
Figure 3.9: Bar Plot of Light Pollution Levels Against Bird Strike Count 150
Figure 3.10: Permutation Importances in Random Forest Model 153
Figure 3.11: Feature Importances in Random Forest Model 153
Figure 3.12: SHAP Values in Random Forest Model (Beeswarm Plot) 154
Figure 4.1: Proposed Framework to Designing Policy in Uncertain Environments 186
Figure 4.2: Integrated Policy Designing Decision Model (IPDDM) 195
Figure 4.3: Coupled Representation and Modeling of Interactions and Interdependencies Among Decisions across Varying Abstraction Levels between Three Decisions 196
Figure 4.4: Formulation of Uncertainty Bounds 198

Figure 4.5: Mathematical Constructs for Design 200
Figure 4.6: Input-Output Component Planes for Test Problem 207
Figure 4.7: Region of Interest Shown by the Red Border Hexagons and Hits Shown by Red Dots 208
Figure 5.1: Conceptual Framework for Goal-Oriented Inverse Design 241
Figure 5.2: Forward and Backward Modeling in Goal-Oriented Inverse Design.....242
Figure 5.3: Historical Trends (2000-2017) and Projected Future Values (2018-2030) 249
Figure 5.4: iSOM Plots for All Weights (Scenarios) 258
Figure 5.5: iSOM Plots for All Goals 258
Figure 5.6: The Black iSOM Grid Points Indicate the Common Satisficing Solution Region for All Goals 258
Figure 5.7: Satisficing Solution Space for State 2023 262
Figure 5.8: iSOM Plot, Updating the Weights 264
Figure 5.9: Sector-Wise Employment Distribution across Scenarios 27 and 28 265
Figure 5.10: Satisficing Solution Space for Transition State Year 2027.....267
Figure 6.1: Proposed Framework Architecture to Support Interpretable Satisficing.....293
Figure 6.2: Knowledge Graph Representation of the Design Goals, Preference Weights, and Hierarchical Levels 308
Figure 6.3: Preliminary iSOM Plots for Goals at Design Levels 1 and 2, where Yellow and Dark Blue Areas Correspond to Regions of Relatively High and Low Robustness, Respectively 312
Figure 6.4: iSOM Plots for All Goals across Design Levels 1 and 2 After Updating the Satisficing Goal Thresholds 315
Figure 7.1: Overview of Ca-Rs-Q.....354
Figure 7.2a: Convergence on Benchmark A. Mean and maximum $\|Q - Q^*\|$ for CA-RS-Q and standard Q-learning.....376
Figure 7.2b: Convergence on Benchmark B over 5,000 episodes 376
Figure 7.3a: Satisficing-rate Trajectory and (G+) Diagnostic on Benchmark A 378
Figure 7.3b: Satisficing-Rate Trajectory and (G+) Diagnostic on Benchmark B.....378
Figure 7.4: Robustness on Benchmark C.....380
Figure 7.5: Post-Shock Catastrophe Rate across the Three Benchmarks on a Logarithmic Axis 383

Figure 7.6: Post-Shock Return Distributions on Benchmark C 384

Figure 7.7: Knowledge-Graph Snapshots over the Spotlight Episode on Benchmark B: Pre-Shock, Mid-Episode, and Late-Episode.....387

Figure 7.8: Per-Step Interpretability Trace over One Benchmark B Episode.....389

Figure 8.1: Overview of the Proposed Cognitive Architecture.....419

Figure 8.2: The Internal Loop of a Single Agent 420

Figure 8.3: The Three Orchestration Designs 427

Figure 8.4: Final Agent Influence and Alignment Network for a Representative Hybrid-Scenario Run 436

Figure 8.5: Stance Trajectories of All Twenty Agents across the Forty-Five-Turn Under the Full Architecture.....437

Figure 8.6: Within-Turn Deliberation Convergence under Design C for Four Representative Turns 438

Figure 8.7: Distribution of Agent Actions across Turns under the Full Architecture 440

Figure 8.8: Growth of the Auditable Change Log over a Single Run (Left), Reaching More than One Thousand Recorded World-State Mutations, with the Breakdown of Those Mutations by Entity Type (Right) 443

Figure 8.9: Knowledge-Graph Schema of the World State: Five Entity Types and Their Principal Relations 444

Figure 8.10: System-Level Dynamics over a Single Run.....445

Figure 8.11: Per-Agent Arousal across the Forty-Five-Turn Horizon (Hybrid Scenario) 446

Figure 8.12: Stance of Every Agent on Every Turn (Hybrid Scenario).....447

Figure 8.13: Storyboard of the Influence Network at Six Representative Turns.....448

Figure 8.14: Directed Information-Flow Matrix Among Stakeholder Classes, Aggregated over the Run 449

Figure 8.15: Complete Cognitive Trace of the Swing Agent (Budget-Conscious Citizen) across the Run 453

Figure 8.16: Complete Cognitive Trace of the Progressive Union across the Run 454

Figure 9.1: Architecture of the Decision State Information Model 481

Figure 9.2: Verification and Validation Square 510

Figure 9.3: Towards Self Designing Systems 546

Figure 9.4: The Two Axes of Progression Converging on the Decision-State Information Model 550

ABSTRACT

Subject: Engineers and decision makers increasingly address systems in which cyber technologies, physical infrastructure, natural environments, human behavior, and institutions evolve together. Engineers developed the classical specify, optimize, and deliver paradigm for systems with relatively stable goals, boundaries, and operating conditions. Engineers face this challenge with particular urgency for two reasons. First, the systems that clients, agencies, and governments ask us to design, among them carbon policy, agriculture, supply chains, and national sustainable development, are increasingly sociotechnical and evolving rather than static and technical. Second, researchers have advanced artificial intelligence rapidly, from generative models and knowledge graphs to autonomous agents and large-scale behavioral simulation, and through those advances they change both the systems we design and the instruments we design with. We address both the complexity that designers must manage and the artificial intelligence and related capabilities that they can use. We treat these problems as instances of a single design object, the Evolving Cyber-Physical-Social System (E-CPSS), and we work within a Strategic Engineering paradigm in which we engage uncertainty and evolution as fundamental features of the design problem rather than treat them as disturbances to suppress. Within this paradigm, our primary goal is to establish the principles, methods, and mathematics a designer uses to engineer robust, adaptive, and socially legitimate interventions across an E-CPSS's state transitions, and to organize them within the Decision-State Information Model, an architecture through which designers develop adaptive decision intelligence, so that designers keep their criteria for a good decision coupled to the changing structure of the system rather than fix those criteria at design time. We address six escalating challenges: (a) reading the tri-space state of an evolving system from heterogeneous signals in a design-interpretable way; (b) structuring coupled, multi-level decisions robustly within a state; (c) using the GOIDM to formulate adaptive strategies by working backward from desired future states across state transitions; (d) augmenting the designer's judgment with AI when designers can no longer navigate the design space unaided; (e) executing strategies autonomously and under governance when the system evolves faster than the human decision cycle; and (f) assessing the social legitimacy of an intervention before we deploy it.

Special features: To address these challenges we develop six methods, one for each of the six escalating design challenges, and we combine them as the five layers of the Decision-State Information Model: observation; robust and adaptive decision structuring; augmented trade-off navigation; governed autonomous execution; and cognitively grounded social-legitimacy assessment. We distinguish the monograph through four features. First, we connect every method to a specific part of a state-indexed design process, ground all six methods in decision-based design and satisficing, and carry validated knowledge from one system state into the next. Second, we arrange the capabilities in an escalation through which designers address increasingly demanding challenges and produce predictive, structural, dynamic, generative, experiential, and simulated knowledge. Third, we demonstrate the methods through six worked examples involving aviation bird strikes, Agriculture 5.0, national sustainable development, steel manufacturing, supply-chain resilience, and carbon taxation. Fourth, we treat interpretability, governance, and social legitimacy as critical design requirements, for example by validating augmentation results against quantitative outcomes before we rely on them, by designing autonomous agents whose decisions we can audit against the facts we used to make them, and by assessing stakeholder acceptance of an intervention before deployment. In each chapter we provide a frame-of-reference section, a worked test problem, and an explicit verification-and-validation discussion anchored in the Validation Square.

Main benefits and contributions: We organize our contributions as six method-level advances of the Scholarship of Discovery and one synthesis-level advance of the Scholarship of Integration. Through the six methods we produce, respectively, predictive knowledge (a CPSS-structured predictive-modeling method for interpretable tri-space state characterization), structural knowledge (a coupled, multi-level robust decision method anchored in the Integrated Policy Designing Decision Model and Design Capability Indices), dynamic knowledge (a Goal-Oriented Inverse Design Method for adaptive strategies across transitions), generative knowledge (a validated generative-AI and knowledge-graph framework for interpretable trade-off navigation), experiential knowledge (a knowledge-governed autonomous-agent architecture with formal guarantees under non-stationarity), and simulated knowledge (a cognitively grounded stakeholder-simulation architecture for pre-deployment legitimacy assessment). Through DSIM, we connect observation, decision structuring, adaptive design, augmented intelligence, governed autonomous execution, and social-legitimacy assessment within a state-indexed architecture for adaptive decision intelligence. We direct this work at a class of problems that researchers and practitioners

in engineering and strategy design increasingly address. The defining challenges of our time are evolving cyber-physical-social systems: they couple technical, physical, and human factors that designers cannot address through any interventions in isolation; the challenges change through the very interventions meant to steer them; their goals shift as they evolve, so that a solution fit for one moment is unfit for the next; and designers face uncertainty that is structural rather than incidental. When designers rely on a single optimize and deploy decision, they may select interventions that no longer fit changed conditions, fail under uncertainty, or encounter stakeholder resistance. Through the methods and the architecture we develop here, a designer can read the state of such a system, structure decisions that are robust and adapt as the system transitions, extend their own judgment when the problem outgrows it, hand execution to governed and auditable agents when the system moves faster than they can act, and test an intervention's legitimacy before committing to it

CHAPTER 1

MOTIVATION AND PROBLEM IDENTIFICATION

1.1 Introduction to the Challenge of Designing Evolving Cyber-Physical-Social Systems

Engineering has always been a discipline of purposeful change. We design systems to achieve goals, and we judge our designs by measuring goal attainment under the conditions we anticipate. Engineers have used this framing effectively when they work with systems whose requirements remain sufficiently stable, whose operating conditions remain reasonably predictable, and whose performance they evaluate against specifications established before deployment. Engineers working within what Rittel and Webber [1] called the first-generation systems approach assumed that they could decompose the world, order objectives, and continue to rely on a solution once they found it. Engineers designed a wide range of systems successfully on that assumption. They can no longer rely on it for a growing class of problems.

When city officials expand the lighting infrastructure, through the physical luminance density we observe reshape in bird migration corridors. Disoriented birds enter commercial flight paths at higher rates, and aviation safety regulators must respond with updated monitoring, airspace management protocols, and reporting requirements. Urban communities, responding to ecological research and policy pressure, begin adopting lighting ordinances, and as they comply with them they alter the physical environment again. We trace this chain as a causally closed loop across three spaces that we distinguish: a physical space of infrastructure, habitat, and biology; a cyber space of FAA reporting systems, geospatial lighting databases, and predictive migration models; and a social space of governance, community behavior, and institutional adaptation. No designer intervening in any one of these spaces produces the intended effect in isolation from the others, and no designer working within any single space can observe the full causal structure from that vantage point. Designers face more than a multidisciplinary problem. It is co-evolving: the spaces do not simply interact, they transform each other over time.

We observe the same structural character when we consider national agricultural transformation. When vendors and extension agencies introduce precision farming technologies in a region, farmers decide whether to adopt them on the basis of the trust they place in the tools, the credit they can obtain, the behavior they observe among their peers, and the subsidies that governments offer. As farmers adopt at scale, they reshape water usage, food security, carbon footprint, and the social conditions under which they first decided to adopt. A designer of Agriculture 5.0 policy who models any one of these pathways without the others will identify leverage points that lose their effect as farmers, agencies, and physical conditions change in response. Policymakers must adapt governance as the physical consequences of adoption reach communities, and the stakeholders in those communities determine which governance measures policymakers can implement. Designers may see a technology adoption problem. At the scale where consequences reach farmers, communities, and policymakers, they face a problem of co-evolving cyber, physical, and social systems.

Policymakers who pursue sustainable national development face a structurally identical challenge over a longer timescale. Policymakers who plan simultaneously for economic growth, energy system transition, and social equity must decide today, and stakeholders, institutions, and physical infrastructure carry the consequences of those decisions over decades. The goals themselves shift, because through the decisions we make earlier we alter the social and economic landscape in which we make the later ones. Policymakers who design a portfolio from today's conditions will find it unfit later, not because they failed to foresee external change, but because they change the conditions themselves when they design and deploy the portfolio. We see the same logic in supply chains at a faster timescale: disruptions reach one actor after another faster than any centralized decision maker can respond; designers program agents that act rationally on local information and thereby destabilize behavior across the network; and the failures that cost stakeholders most arise not because a single component broke, but because no one designed the system to be managed as an evolving, coupled sociotechnical entity.

These are not four different problems. They are four manifestations of the same design object. In each example, people make decisions within interconnected cyber, physical, and social conditions, and through those decisions they alter the conditions under which they decide later. We use the term Evolving Cyber Physical Social System to describe this shared design situation. An E-CPSS

is a system in which a cyber space of computational platforms, data infrastructure, and algorithmic agents; a physical space of engineered artifacts, ecological components, and measurable phenomena; and a social space of human cognition, behavioral dynamics, institutional structures, and governance frameworks co-evolve through reciprocal causal coupling across time. Through the cognitive map of an E-CPSS (as described in detail in Chapter 2) we illustrate how the boundaries between these spaces are not fixed partitions but dynamic interfaces through which changes in one space alter the conditions of design and decision-making in the others. The CPSS evolves not through external disruption alone, but through the endogenous consequences of the decisions made within it.

The idea that engineered systems should be treated as co-evolving rather than static is not new to systems theory, but it has remained at the periphery of engineering design practice. Allen and co-authors [2] observe that designing evolving CPSS requires making tradeoffs simultaneously across cyber, physical, and social systems, and that through advances in computation designers can now ask the difficult and consequential questions they encounter in this class of problem. Sobb and co-authors [3], reviewing the CPSS literature broadly, find that the field remains comparatively nascent: CPSS research comprises roughly 500 active publications against more than 25,000 for conventional CPS, and researchers have not yet provided explicit guidance through which a designer unifies evolvability with systematic decision support across all three spaces. We do not identify this gap solely through a bibliometric observation. We interpret it as evidence that we have not yet developed the capacity to engineer systems that change as we act on them.

Researchers working on flexibility and real options in complex systems increasingly attend to the temporal dimension of engineering design. Cardin [4] develops a taxonomy of procedures for enabling flexibility in the design and management of engineering systems under uncertainty, identifying five phases through which designers can systematically prepare a system to respond to futures they cannot fully anticipate. Through this work, we contribute to the literature on designing for change. These authors do not address the case that designers meet in E-CPSS design, in which the people, technologies, and physical conditions of the three spaces change one another and in which designers find the relevant information, decisions, methods, and knowledge to be state-dependent and temporally indexed. The uncertainty in an E-CPSS is not exogenous and stationary; it is produced by the system's own evolution.

The deeper difficulty is what Rittel and Webber [1] named five decades ago when they observed that the classical paradigm of science and engineering is not applicable to the problems of open societal systems. They argued that when stakeholders hold conflicting values, when no one can definitively specify the requirements, and when deploying a solution alters the conditions of the problem, engineers cannot address the problem through the scientific-engineering model of diagnosis and solution. Designers meet these characteristics in E-CPSS design problems: the cyber, physical, and social spaces are coupled in ways that make unilateral optimization of any single objective both technically incomplete and socially illegitimate. The designer of an E-CPSS must engage with genuine co-evolutionary complexity, and must do so in a manner that is robust to the uncertainty of the system's own trajectory.

To address this challenge, we must make a paradigmatic commitment rather than merely assemble a collection of new methods. Simon's (1955) insight that bounded rationality and satisficing, the pursuit of solutions that are good enough under realistically achievable conditions rather than globally optimal under idealized ones, is the correct decision orientation for complex, incompletely specified problems provides the philosophical foundation. Jiao and co-authors [5] identify the emerging frontier of Design Engineering in the age of Industry 4.0 and beyond as one that requires a human-cyber-physical view of the systems realization ecosystem, integrating the cognitive and behavioral dimensions of sociotechnical systems into the design process itself. The paradigm we work within, which we call Strategic Engineering, takes this orientation further: we design and manage technological systems to create and maintain value across uncertain, evolving futures, treating the temporal dimension of system evolution not as a complication to be controlled but as a fundamental feature of the design problem to be engaged.

In this dissertation, we develop the methods and architecture for doing exactly this. In the sections that follow, we frame Strategic Engineering as the paradigmatic commitment that orients the work, identify two fundamental asymmetries that characterize the E-CPSS design challenge, present six escalating design challenges through which those asymmetries manifest across our five test contexts, and introduce the Decision-State Information Model as the synthesis target: the framework through which a designer establishes, at every state of an E-CPSS, what information is available, what decisions must be made, what methods are applicable, what knowledge is

produced, and what must be true before the system can legitimately advance. In Figure 1.1, we present the overall escalation arc of the dissertation.

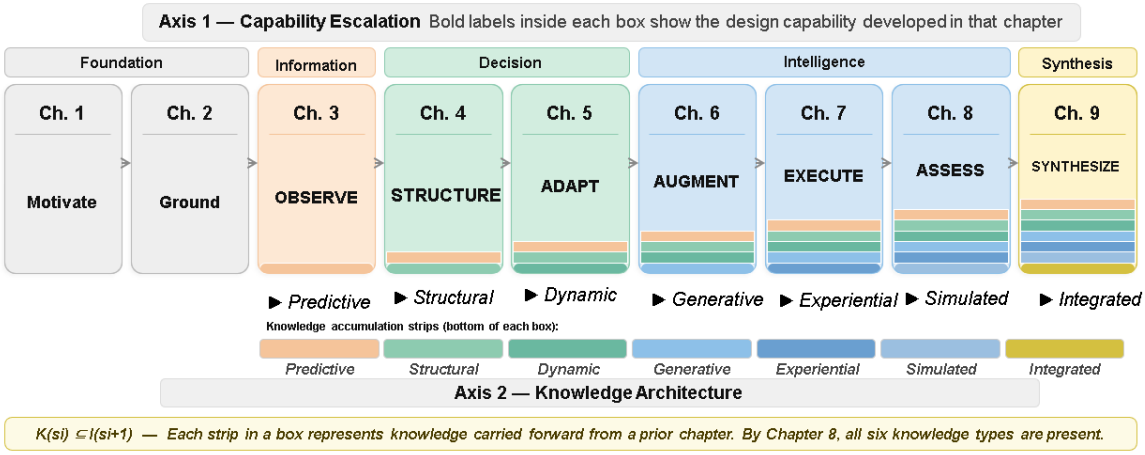


Figure 1.1: Dissertation Progression

1.2 Strategic Engineering as the Paradigmatic Frame

We need a paradigm for what we are doing before we need methods for how to do it. Through a method we answer the question of how to solve a specific class of design problem given specified inputs and constraints. Through a paradigm we answer the question of what kind of designer one must be, and what kind of thinking one must engage in, to navigate problems whose structure itself evolves.

Strategic Engineering is the design and management of technological systems to create and maintain value across uncertain, evolving futures. We use the word strategic deliberately. In strategic board games such as chess or Go, the goal is not to find the optimal response to the current position. The player manages the game so as to preserve and expand optionality, recognizes when the nature of the problem has shifted, and stays positioned to make sound decisions across a range of future conditions that no one can fully anticipate. A skilled chess player understands the positional logic of the game well enough to recognize which features of the current position matter, what future states are accessible, and through which moves the player expands rather than forecloses the space of desirable outcomes. Strategic thinking is inherently about time, uncertainty, and the structure of the decision space rather than the optimization of a fixed objective.

To engineer an E-CPSS a designer must think in exactly this way, and for exactly the reasons that make it insufficient to treat E-CPSS design as constrained optimization from a fixed position. The system is always in motion. The information available to us changes as it evolves. The objectives relevant at one state may not be the objectives relevant at the next. The interventions effective within a state may not be the interventions through which a designer guides the system soundly through a transition. The criteria by which stakeholders recognize success are themselves partially social and subject to renegotiation as the stakeholder landscape shifts. Under these conditions, designers working within the classical paradigm, in which we specify, optimize, and deliver, do not fail gracefully. They produce interventions that are technically precise and contextually obsolete, because the context they optimized for continues to evolve after they deploy them.

Within the paradigm of Strategic Engineering, we do not ask whether a globally optimal solution to a fully specified problem exists. We ask how to design a system through which a designer continues to pursue goals robustly across a horizon of evolving environments. We treat uncertainty as a fundamental feature of the design problem and build methods through which a designer meets the performance standards they set under uncertainty rather than despite it.. We adopt the satisficing orientation from the decision based design tradition because designers rarely possess complete information, fixed preferences, or stable operating conditions in an E CPSS. In adopting it we recognize that in high-dimensional, multi-stakeholder, evolving systems the notion of a single optimal solution is rarely coherent, and that designers searching for one often produce designs that fail when conditions shift by amounts no planner could have anticipated.

We develop every method in this dissertation by applying Strategic Engineering thinking to a specific E-CPSS design challenge. In Chapter 3, we honor our commitment, as strategic engineers, to reading the system's state before acting on it. In Chapter 4, we hold to our insistence on structuring decisions explicitly and robustly before committing to them. In Chapter 5, we act on our preference for working backward from desired future states rather than forward from current conditions. In Chapter 6, we act on our recognition that to maintain human agency in the face of high-dimensional complexity we must augment cognition rather than replace it. In Chapter 7, we honor our commitment to executing strategies at the timescale of the system rather than the human decision cycle. In Chapter 8, we act on our demand that we treat social legitimacy as a design requirement and not as an afterthought.

We unify these six contributions through the Decision-State Information Model, in which we make Strategic Engineering architecturally explicit. Through the DSIM we define what a designer must know at each state of an E-CPSS, what decisions they must make, which methods they can apply, what knowledge they produce and carry forward, and what must be true before they advance the system legitimately to the next state.

1.3 Two Fundamental Asymmetries in E-CPSS Design

We trace every specific design challenge in an E-CPSS, regardless of domain or scale, back to one or both of two structural difficulties. We call these information asymmetry and intelligence asymmetry. We identify them precisely because we do not frame our research questions simply as responses to empirical problems in specific domains. They are responses to these two asymmetries as they manifest across a range of E-CPSS design situations. From the two together we explain why we need a multi-capability framework rather than a collection of independent methods.

1.3.1 Information Asymmetry

An E-CPSS continuously produces signals from all three of its spaces. Aviation authorities use aviation monitoring systems to generate strike incident records with timestamps, coordinates, aircraft types, and environmental conditions. Light pollution GIS platforms map artificial luminance density across the United States. BirdCast migration models estimate the density and flight bearing of migratory bird populations on any given night. Taken individually, we cannot account for the full phenomenon of interest through any one of these datasets: the causal relationship between artificial lighting, avian behavior, flight corridor exposure, and strike incidence. Even when we combine them, we obtain a collection of correlated variables rather than a coherent characterization of the system state, unless we integrate them through a principled approach.

Information asymmetry is the condition in which the system produces more heterogeneous signals than we can organize into actionable intelligence. We are data-rich and interpretation-poor. Designers do not struggle because they lack data. They struggle because they lack a structuring principle through which we map available data onto the tri-space representation of the E-CPSS in support of our design decisions. Without such a principle, we either reduce the data to the variables

most familiar from prior work, discarding the cross-space causal structure, or treat it as an undifferentiated feature pool and extract statistical patterns that are accurate but uninterpretable for design purposes.

In Chapter 3, we address information asymmetry by developing a CPSS-structured predictive modeling method through which we organize observational features according to the tri-space structure of the E-CPSS before we construct the model. Through this method, we construct the first formal component of the Decision-State Information Model, which we denote I(si): the structured tri-space characterization of the E-CPSS at state si which we carry into every subsequent layer of the framework. Designers do not eliminate information asymmetry once we establish the observation framework. They meet it again at every state transition, because the system presents different signals at si+1 than at si. Designers therefore orient themselves continually toward structured observation at each state, and we formalize that orientation through the I(si) component and the transition condition T(si to si+1).

1.3.2 Intelligence Asymmetry

Even when information asymmetry is resolved, a second structural difficulty remains. The design problems that emerge from a well-characterized E-CPSS state are often high-dimensional, multi-objective, and coupled in ways that exceed the cognitive tractability of human designers working with conventional decision support tools. When stakeholders ask us to specify a policy portfolio through which we pursue goals of economic productivity, environmental sustainability, food security, and social equity simultaneously across multiple stakeholder groups with heterogeneous preferences, under uncertainty about adoption dynamics and over a planning horizon during which the goals themselves may shift, the solution space is not merely large. Interactions and trade-offs structure that space, and we cannot navigate them through intuition or by examining low-dimensional projections of the design space.

Intelligence asymmetry is the condition in which the analytical and cognitive capacity available to us is insufficient for the decisions we must make in the E-CPSS design problem. The asymmetry is structural: it grows with the complexity of the E-CPSS state, not with the capability of any particular designer. As an E-CPSS moves through states of increasing complexity, designers must bring more analytical capacity to each successive design problem. We cannot resolve intelligence

asymmetry by applying a more powerful optimizer to the same problem formulation, because through more powerful optimization we do not close the interpretability gap: a solution we cannot explain to the people it affects is not a valid solution in the Strategic Engineering sense.

What we need is a capability architecture through which we augment our intelligence with interpretable computational assistance and, where the system evolves faster than we can respond, with autonomous execution that carries our structured intent without requiring our presence at every decision point. Across Chapters 4 through 7, we address intelligence asymmetry through a sequence of methods that we make progressively more capable as the E-CPSS state grows more complex.

We emphasize that resolving one asymmetry without the other is insufficient. A designer who resolves information asymmetry but not intelligence asymmetry holds an accurate picture of the system state with no method for acting on it. A designer who resolves intelligence asymmetry but not information asymmetry holds powerful methods and searches with them in the wrong space. We express this joint commitment in both the PRQ and the DSIM: we must be capable of seeing the system clearly and reasoning about it well, simultaneously and across every state of the E-CPSS lifecycle.

1.4 Six Escalating Challenges in E-CPSS Design

Through the two asymmetries we identify the structural source of difficulty in E-CPSS design. In this section, we identify six specific design challenges through which those asymmetries manifest as practical problems across the dissertation's test contexts. We escalate these challenges: through each we build on the difficulties of the preceding ones and expose a new dimension of the E-CPSS design problem that we do not expose through the earlier ones. We present them in the order in which we address them in this dissertation, naming for each the test context in which we see the challenge most sharply.

The first challenge is observation. Before we design any intervention, we must characterize the current state of the E-CPSS across all three spaces simultaneously. For this we need more than access to data; we need a principled approach to structuring heterogeneous signals into a coherent, design-relevant state representation. Without it, we operate from an impoverished picture of the

system and cannot construct interventions sensitive to the full causal structure. In Chapter 3, we address the observation challenge through the CPSS-structured predictive modeling method, which we demonstrate on the artificial light at night and bird strikes test problem.

The second challenge is decision structuring. Once we characterize the state, we face the task of organizing the decisions we must make within it. In an E-CPSS, decisions are not flat; we make them at multiple levels of abstraction simultaneously. Policymakers set national agricultural policy at the macro level, agencies run implementation programs at the regional meso level, and an individual farmer decides at the micro level whether to adopt. Stakeholders at these levels interact: through their macro-level choices policymakers shape the incentives under which agencies implement, and through those incentives agencies shape the conditions under which farmers decide. A designer who treats all decisions as equivalent, or who addresses only one level, produces interventions that are internally inconsistent or that leave coupling effects unaddressed. In Chapter 4, we address the decision structuring challenge through the Integrated Policy Designing Decision Model, demonstrated on the Agriculture 5.0 test problem.

The third challenge is adaptability across state transitions. Robust design within a state is necessary but not sufficient. An E-CPSS moves through states, and a designer must typically redesign an intervention effective within a state as the system transitions. The challenge is to design not only a sound intervention for the current state but a principled process through which we derive the next intervention from the outcomes we observe and the knowledge we produce at the current state. For this a designer works backward from desired future states rather than forward from current conditions, and explicitly manages the uncertainty that arises at state transitions. In Chapter 5, we address the adaptability challenge through the Goal-Oriented Inverse Design Method, demonstrated on the UAE sustainable development test problem.

The fourth challenge is cognitive tractability. As the complexity of the E-CPSS grows, the design problem grows with it. When the number of goals, the dimensionality of the decision space, and the density of inter-goal trade-offs exceed the bounds of human interpretability, standard decision support tools produce outputs we cannot meaningfully navigate. The challenge for designers is to augment human judgment without abandoning it: to obtain interpretable, queryable computational guidance through which we expand our effective cognitive reach without surrendering our agency or our responsibility. In Chapter 6, we address this challenge through the Generative AI and

Knowledge Graph framework, demonstrated on the hot rod rolling steel manufacturing test problem.

The fifth challenge is autonomous execution. Some E-CPSS evolve at a pace the human design cycle cannot match. Supply chain disruptions propagate through global networks in hours. As real-time demand signals arrive, designers must revise their decisions within minutes. The challenge is to deploy agents capable of executing adaptive strategies autonomously, sharing structured knowledge across the agent network, and remaining within the governance constraints we specify as designers, all without requiring human intervention at every step. In Chapter 7, we address this challenge through two complementary autonomous agent architectures, demonstrated on the supply chain resilience test problem.

The sixth challenge is social legitimacy. We cannot count an intervention as a successful E-CPSS design, however well it performs against the technical measures we set, if the stakeholders whom it affects reject, resist, or subvert it. Social legitimacy is not a soft consideration a designer appends to a technical process. It is a design requirement with measurable implications for whether a designer achieves the intended effect through an intervention. The challenge is to evaluate social legitimacy before deployment by modeling how stakeholder populations respond, through cognitively grounded agent simulation in which we capture the part memory, emotion, and institutional constraint play as stakeholders accept or resist. In Chapter 8, we address this challenge through the memory-emotion coupled stakeholder agent architecture, demonstrated on the carbon emission taxation test problem.

In Table 1.1, we present the cross-reference of research questions to dissertation chapters, showing in which chapters we provide the foundation, justification, and validation evidence for each research question, and how we map six escalating challenges to the layers of the Decision-State Information Model.

Table 1.1: Research Questions to Chapters and Validation Quadrants

F = Foundation, J = Justification, Q1 = Theoretical Structural Validity, Q2 = Empirical Structural Validity, Q3 = Empirical Performance Validity, Q4 = Theoretical Performance Validity, S = Synthesis and Closure.
SoI = Scholarship of Integration, SoD = Scholarship of Discovery

Research Question	TP	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	Ch 7	Ch 8	Ch 9
PRQ (SoI) What are the principles and methods that designers can use to strategically engineer robust, adaptive, and intelligent interventions in Evolving Cyber-Physical-Social Systems across state transitions under uncertainty?	All	F	F							Q4, S
Hypothesis: Through the progressive development and integration of principled methods for state observation, robust and adaptive decision structuring, augmented intelligence, governed autonomous execution, and social legitimacy assessment, designers can strategically engineer interventions in Evolving Cyber-Physical-Social Systems that are robust under uncertainty, adaptive across state transitions, and socially legitimate before deployment. **New Knowledge:** The Decision-State Information Model: a state-indexed design intelligence framework for the strategic engineering of Evolving Cyber-Physical-Social Systems.										
RQ1 / H1 (SoD) What are the principles and the method for leveraging the cyber-physical-	ALAN	F	J	Q1, Q2, Q3						Q4, S

social structure of an E-CPSS in predictive modelling to uncover contributing factors and support design decisions?					
Hypothesis 1: Through a CPSS-structured feature organization that assigns observational variables to cyber, physical, and social spaces before model training, and SHAP-based importance attribution mapped back to those spaces after training, designers construct interpretable tri-space state characterizations that reveal the contribution of ALAN to bird strikes and yield actionable insights for public policy and aviation safety decision-making. **New Knowledge 1:** A CPSS-structured predictive modelling method for constructing interpretable tri-space state characterizations of Evolving Cyber-Physical-Social Systems to support design decisions. Empirical confirmation that Artificial Light at Night is a statistically significant contributing factor to bird strikes, with light pollution level identified as the strongest predictor, with implications for public policy and aviation safety.					
RQ2a / H2a (SoD) What is the mathematics and method for designing robust strategies in an E-CPSS through coupled, multi-level decisions under uncertainty?	Agriculture 5.0	F	J	Q1, Q2, Q3	Q4, S
Hypothesis 2a: Through modeling E-CPSS interventions as coupled compromise Decision Support Problems that encode decision types, interaction strengths, and hierarchical levels					

using the Integrated Policy Designing Decision Model, with Design Capability Index constraints enforcing robustness explicitly, designers produce satisficing solutions that remain robust across stakeholder priority configurations under uncertainty. **New Knowledge 2a :** A coupled decision method anchored in the Integrated Policy Designing Decision Model and Design Capability Indices for designing robust multi-level satisficing strategies in Evolving Cyber-Physical-Social Systems under uncertainty.					
RQ2b / H2b (SoD) What are the principles, mathematics, and method for designing adaptive strategies from desired future states of an E-CPSS under uncertainty?	UAE Sus-Dev	F	J	Q1, Q2, Q3	Q4, S
Hypothesis 2b: By using the Goal-Oriented Inverse Design Method to work backward from desired future states, enforce Design Capability Index constraints at every state transition, and update goals through feedback when observed performance deviates from projections, designers design adaptive strategies that remain robust across E-CPSS state transitions under uncertainty. **New Knowledge 2b :** A Goal-Oriented Inverse Design Method anchored in the principle of working backward from desired future states, with DCI-enforced transition conditions and feedback-driven goal updating, for deriving adaptive strategies across E-CPSS state transitions under uncertainty.					
RQ3a / H3a (SoD) What are the principles and method a designer	HRR steel MFG	F	J	Q1, Q2, Q3	Q4, S

uses to enable interpretable and adaptive trade-off navigation when E-CPSS design problems exceed human cognitive tractability?					
Hypothesis 3a: Through a generative AI framework in which an LLM extracts knowledge statements from structured scenario data, each statement is validated quantitatively before knowledge graph ingestion, and the validated graph supports interactive querying and adaptive infeasibility recovery, designers navigate high-dimensional satisficing spaces with interpretable and actionable guidance for design decision-making. **New Knowledge 3a:** A GenAI-anchored Knowledge Graph framework for augmenting E-CPSS design decision-making, with LLM-based knowledge extraction, quantitative validation protocols, and interactive trade-off summarization for interpretable guidance in high-dimensional design spaces.					
RQ3b / H3b (SoD) What are the principles and method for enabling an autonomous agent to execute adaptive strategies in an E-CPSS under non-stationary conditions, while preserving the robustness, adaptivity, and	SCM	F	J	Q1, Q2, Q3*	Q4, S

interpretability established in the preceding chapters?					
Hypothesis 3b: An agent that draws its standard for acceptable performance from a structured, dynamically updated description of the world maintains satisficing performance and adapts more reliably under non-stationary E-CPSS conditions than an agent that learns from the reward signal alone, without surrendering the convergence, before-convergence, robustness, and auditability guarantees a high-stakes setting requires. **New Knowledge 3b:** A method for sourcing a reinforcement-learning agent's satisficing aspiration from a structured, dynamically updated Knowledge Graph, the Compiled-Aspiration MDP and the CA-RS-Q algorithm which keeps value learning and aspiration formation in separate channels, carries convergence, finite-time satisficing-rate, and graph-perturbation robustness guarantees, and renders each decision auditable to the facts that set the aspiration, for autonomous strategic execution in non-stationary E-CPSS.					
RQ4 / H4 (SoD) What are the principles and method for modeling stakeholder behavior through cognitively grounded agent simulation to support pre-deployment social legitimacy assessment of E-CPSS interventions?	Carbon emission taxation	F	J	Q1, Q2, Q3*	Q4, S
Hypothesis: By designing stakeholder agents whose emotional influence on cognition is grounded in predictive-processing principles, specifically by treating arousal as a proxy for					

epistemic uncertainty that dynamically scales how strongly emotional state biases memory retrieval, and by grounding all agents in a shared knowledge graph with role constraints enforced through a rule system, designers produce behavioral simulations that are measurably distinct from rational-actor models and yield diagnostic pre-deployment information about stakeholder resistance before any intervention is deployed.

New Knowledge: A cognitively grounded stakeholder agent architecture with predictive-processing-anchored memory-emotion coupling, neuro-symbolic knowledge graph grounding, and automated agent generation for pre-deployment social legitimacy assessment of adaptive interventions in Evolving Cyber-Physical-Social Systems.

1.5 The Decision-State Information Model as the Synthesis Target

We have established the design object, the paradigm, the structural source of difficulty, and the specific challenges in which designers encounter that difficulty. We turn now to the synthesis target of this dissertation: the Decision-State Information Model.

Through the DSIM, we answer a question that we cannot avoid once we consider the six escalating challenges together. Across every state of an E-CPSS, we simultaneously face three questions. What do we know about the system right now? What decisions should we make given that information, and what methods are applicable? What will we know after we have acted, and how should we carry that knowledge forward to inform the next state? Working within static design frameworks, designers answer these questions for systems in which the design problem does not change. An E-CPSS evolves, and the design problem changes with each state. The DSIM is our response to this: the framework through which we establish, at state s_i of an E-CPSS, exactly what information is available and how to structure it, what decisions are warranted and at what level of abstraction, what methods are appropriate given the complexity of those decisions, what knowledge is produced, and what must be true before the system can legitimately transition to s_{i+1}.

We define the DSIM formally as the six-tuple {S, I, D, M, K, T}. S is the ordered set of states through which the E-CPSS evolves; in the UAE test problem, S encompasses the current state

characterized by historical data through 2017, a transition state for 2023, and the target state for 2030. I(si) is the structured tri-space characterization at state si, which we construct through the CPSS-structured modeling method of Chapter 3; through I(si), we organize cyber, physical, and social signals into a coherent, design-relevant state representation. D(si) and D(si to si+1) are the intra-state and inter-state decision maps, which we construct through the IPDDM of Chapter 4 and the GOIDM of Chapter 5, respectively. M(si) is the set of methods we activate at state si, escalating from structured decision support to generative AI augmentation to autonomous agent execution as state complexity grows. K(si) is the knowledge we produce at state si and carry forward; DCI values, Knowledge Graph heuristics, reinforcement learning experience, and social legitimacy assessment are all instances of K(si), each tagged to the state at which they were produced and fed into I(si+1) to create the learning loop that distinguishes the DSIM from a collection of static methods. T(si to si+1) is the transition condition: four simultaneous criteria, DCI robustness above threshold, Knowledge Graph updated with validated heuristics, agent aspiration satisfaction confirmed, and social legitimacy score above threshold, that must all hold before we can authorize the advance from si to si+1.

We should be clear about one point concerning the provenance of the DSIM. We did not pre-specify the DSIM and then test it. We developed six methods in response to design challenges that designers meet in E-CPSS across six domains, and in Chapter 9 we demonstrate that these methods, examined together, constitute a coherent and sufficient framework formally described as {S, I, D, M, K, T}. The DSIM is a retrospective synthesis, not a prospective template. On this basis we situate our contribution within Boyer's Scholarship of Discovery and Scholarship of Integration. In Chapters 3 through 8, we contribute through discovery: in each we produce new methods, mathematics, or architectures. In Chapter 9, we contribute through integration: we demonstrate that the discoveries together constitute an architecture for Strategic Engineering of evolving sociotechnical systems.

In Figure 1.2, we present a schematic of the DSIM showing the five operational layers, the six formal components, and the feedback loop through which we carry K(si) forward into I(si+1). We annotate the transition condition T at the arc connecting the Social Assessment Layer output at si to the observation step at si+1.

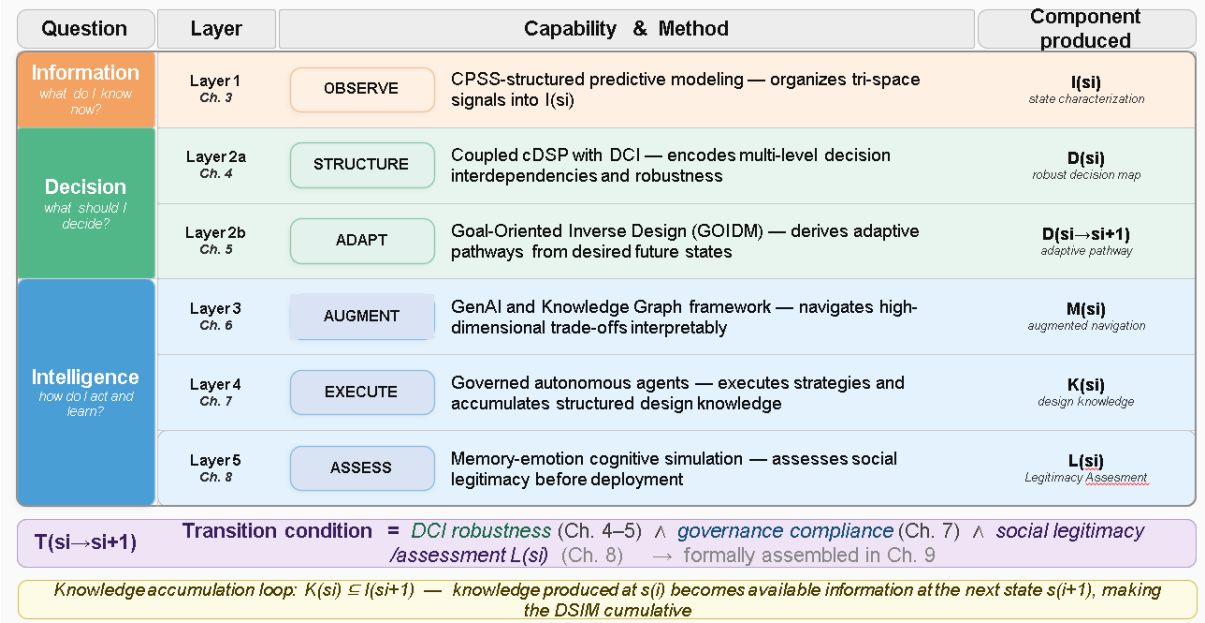


Figure 1.2: Decision-State Information Model

1.6 Research Questions and Hypotheses

Through every research question in a dissertation we simultaneously claim something about what is unknown and commit ourselves to a way of investigating it. Before we present the specific questions we address in this dissertation, we should be deliberate about the philosophy on which we have framed them, and about the logical chain through which we connect the gap analysis of the preceding sections to the questions we ask.

We rest the motivational argument we develop in Sections 1.1 through 1.5 on three cumulative observations. First, designers increasingly encounter E-CPSS as a class of design object that, because of its co-evolutionary character, they cannot address through the tools and assumptions of classical engineering design. Second, two structural asymmetries, an information asymmetry between the heterogeneous signals produced through the system and our capacity to organize them as a coherent state characterization, and an intelligence asymmetry between the cognitive and analytical capacity available to us and the complexity of the decisions we must make within the system, run through every E-CPSS design challenge regardless of domain. Third, designers encounter these asymmetries as six escalating design challenges, and for each challenge designers

need a principled method that no one has yet developed in the integrated, state-indexed form we require for the DSIM.

A research question, in the sense of Boyer's Scholarship of Discovery, is a question we answer by producing knowledge that did not exist before. We ask each secondary research question because there is a specific capability that an E-CPSS designer requires and that we develop and validate through principled method development. In answering each question we produce a precisely stated piece of new knowledge. We frame the Principal Research Question through Boyer's Scholarship of Integration. We ask what the answers to the secondary research questions mean when we consider them together. To answer the Principal Research Question, we must demonstrate how the individual contributions connect and why those connections constitute a coherent architecture rather than a collection of independent methods.

We have framed these questions under several fundamental assumptions, and we state them explicitly. We assume that designers pursue goals when they design an E-CPSS under uncertainty, meaning that we can specify meaningful goals even when we cannot know in advance every condition under which we pursue them. We do not assume that goals stay fixed. We assume that designers and stakeholders update them through feedback as conditions change, and that we must build methods through which they carry out that updating rather than methods that prevent it. We assume that satisficing is the appropriate decision criterion for E-CPSS design: we seek solutions that are good enough across the range of conditions they expect, and we use Design Capability Indices as the mathematical vehicle for making this criterion rigorous and enforceable. We assume that the knowledge we produce at each state of an E-CPSS has value for the decisions we make later, and that designers should carry it forward rather than discard it. And we assume that social legitimacy is a design requirement, not an evaluation criterion applied after the technical design is complete.

From these commitments, and from the argument of Sections 1.1 through 1.5, we derive the Principal Research Question and four secondary research questions. RQ2 consists of two sub-questions because within the Decision Layer of the DSIM a designer carries two distinct responsibilities: structuring decisions robustly within a state through $D(s_i)$, and structuring them adaptively across state transitions through $D(s_i \text{ to } s_{i+1})$. We divide RQ3 into two sub-questions because through the Augmentation and Execution Layers we address related but distinct design

conditions: we augment human intelligence when the dimensionality of the problem exceeds cognitive tractability, and we replace human execution speed when the E-CPSS evolves faster than a designer can deliberate. We address each sub-question through a distinct method and hypothesis, and we validate each independently on an appropriate test problem.

We present the questions, hypotheses, and expected new knowledge in the order in which we develop them, organized by the DSIM layer we address through each. In Table 1.1 we map every research question to the chapters in which we establish its foundation, justify the gap, validate the method, and close the argument through synthesis.

Principal Research Question (PRQ) *(Scholarship of Integration)* What are the principles and methods that designers can use to strategically engineer robust, adaptive, and intelligent interventions in Evolving Cyber-Physical-Social Systems across state transitions under uncertainty?

Hypothesis (PRQ). Through the progressive development and integration of principled methods for state observation, robust and adaptive decision structuring, augmented intelligence, governed autonomous execution, and social legitimacy assessment, designers can strategically engineer interventions in Evolving Cyber-Physical-Social Systems that are robust under uncertainty, adaptive across state transitions, and socially legitimate before deployment.

New Knowledge (PRQ). The Decision-State Information Model: a state-indexed design intelligence framework for the strategic engineering of Evolving Cyber-Physical-Social Systems.

We address the PRQ in Chapter 9 through the formal assembly and completeness argument of the DSIM. Through the secondary research questions we address the component contributions from which we construct the DSIM, and we ground each in the Scholarship of Discovery.

RQ1 *(DSIM Layer 1 — Observation)* What are the principles and the method for leveraging the cyber-physical-social structure of an E-CPSS in predictive modeling to uncover contributing factors and support design decisions?

Gap addressed. We develop a method for organizing observational features according to the tri-space structure of an E-CPSS before model construction, building a predictive model on this structured representation, and interpreting outputs through attribution mapped back to the three spaces, such that the model outputs are interpretable as design-relevant state characterizations.

Hypothesis 1. Through a CPSS-structured feature organization that assigns observational variables to cyber, physical, and social spaces before model training, and SHAP-based importance attribution mapped back to those spaces after training, we construct interpretable tri-space state characterizations that reveal the contribution of ALAN to bird strikes and yield actionable insights for public policy and aviation safety decision-making.

New Knowledge 1. A CPSS-structured predictive modeling method for constructing interpretable tri-space state characterizations of Evolving Cyber-Physical-Social Systems to support design decisions. Empirical confirmation that Artificial Light at Night is a statistically significant contributing factor to bird strikes, with light pollution level identified as the strongest predictor, with implications for public policy and aviation safety.

RQ2a *(DSIM Layer 2, intra-state — Decision Structuring)* What is the mathematics and method for designing robust strategies in an E-CPSS through coupled, multi-level decisions under uncertainty?

Gap addressed. We develop a method for formulating multi-level E-CPSS interventions as coupled decision problems that explicitly represent decision types, interaction strengths, and hierarchical levels, while enforcing robustness across stakeholder priority configurations through a mathematically grounded constraint.

Hypothesis 2a. By modeling E-CPSS interventions as coupled compromise Decision Support Problems that encode decision types, interaction strengths, and hierarchical levels using the Integrated Policy Designing Decision Model, with Design Capability Index constraints enforcing robustness explicitly, we produce satisficing solutions that remain robust across stakeholder priority configurations under uncertainty.

New Knowledge 2a. A coupled decision method anchored in the Integrated Policy Designing Decision Model and Design Capability Indices for designing robust multi-level satisficing strategies in Evolving Cyber-Physical-Social Systems under uncertainty.

RQ2b *(DSIM Layer 2, inter-state — Adaptive Design)* What are the principles, mathematics, and method for designing adaptive strategies from desired future states of an E-CPSS under uncertainty?

Gap addressed. We develop a method for working backward from desired future states of an E-CPSS to derive the interventions through which designers reach those states, while enforcing DCI-based robustness at every state transition and updating goals when observed performance deviates from projections.

Hypothesis 2b. By using the Goal-Oriented Inverse Design Method to work backward from desired future states, enforce Design Capability Index constraints at every state transition, and update goals through feedback when observed performance deviates from projections, we produce adaptive strategies that remain robust across E-CPSS state transitions under uncertainty.

New Knowledge 2b. A Goal-Oriented Inverse Design Method anchored in the principle of working backward from desired future states, with DCI-enforced transition conditions and feedback-driven goal updating, for deriving adaptive strategies across E-CPSS state transitions under uncertainty.

RQ3a *(DSIM Layer 3 — Augmented Intelligence)* What are the principles and method a designer uses to enable interpretable and adaptive trade-off navigation when E-CPSS design problems exceed human cognitive tractability?

Gap addressed. We develop a method for extracting validated design heuristics from structured E-CPSS scenario data, ingesting them into a query able knowledge structure, and delivering interpretable, intent-aware guidance to a designer navigating a high-dimensional satisficing space.

Hypothesis 3a. By deploying a generative AI framework in which a large language model extracts knowledge statements from structured scenario data, each statement is validated quantitatively before Knowledge Graph ingestion, and the validated graph supports interactive querying and adaptive infeasibility recovery, we navigate high-dimensional satisficing spaces with interpretable and actionable guidance for design decision-making.

New Knowledge 3a. A GenAI-anchored Knowledge Graph framework for augmenting E-CPSS design decision-making, with LLM-based knowledge extraction, quantitative validation protocols, and interactive trade-off summarization for interpretable guidance in high-dimensional design spaces.

RQ3b *(DSIM Layer 4 — Autonomous Execution)* What are the principles and method for enabling an autonomous agent to execute adaptive strategies in an E-CPSS under non-stationary conditions, while preserving the robustness, adaptivity, and interpretability established in the preceding chapters?

Gap addressed. We develop architecture for autonomous multi-agent systems in E-CPSS where we combine the retrieval of validated design heuristics from a knowledge structure with policy learning, while remaining bounded by governance constraints specified by the designer.

Hypothesis 3b: An agent that draws its standard for acceptable performance from a structured, dynamically updated description of the world maintains satisficing performance and adapts more reliably under non-stationary E-CPSS conditions than an agent that learns from the reward signal alone, without surrendering the convergence, before-convergence, robustness, and auditability guarantees a high-stakes setting requires.

New Knowledge 3b: A method for sourcing a reinforcement-learning agent's satisficing aspiration from a structured, dynamically updated Knowledge Graph, the Compiled-Aspiration MDP and the CA-RS-Q algorithm which keeps value learning and aspiration formation in separate channels, carries convergence, finite-time satisficing-rate, and graph-perturbation robustness guarantees, and renders each decision auditable to the facts that set the aspiration, for autonomous strategic execution in non-stationary E-CPSS.

RQ4 *(DSIM Layer 5 — Social Assessment)* What are the principles and method for modeling stakeholder behavior through cognitively grounded agent simulation to support pre-deployment social legitimacy assessment of E-CPSS interventions?

Gap addressed. We develop a method for simulating stakeholder behavioral response to E-CPSS interventions through agents whose cognition is grounded in predictive-processing principles, producing pre-deployment legitimacy assessments from which designers diagnose the sources of stakeholder resistance before they commit to an intervention.

Hypothesis 4. By designing stakeholder agents whose emotional influence on cognition is grounded in predictive-processing principles, specifically by treating arousal as a proxy for epistemic uncertainty that dynamically scales how strongly emotional state biases memory retrieval, and by grounding all agents in a shared knowledge graph with role constraints enforced through a rule system, we produce behavioral simulations that are measurably distinct from rational-actor models and yield diagnostic pre-deployment information about stakeholder resistance before any intervention is deployed.

New Knowledge 4. A cognitively grounded stakeholder agent architecture with predictive-processing-anchored memory-emotion coupling, neurosymbolic knowledge graph grounding, and automated agent generation for pre-deployment social legitimacy assessment of adaptive interventions in Evolving Cyber-Physical-Social Systems.

1.7 Two Perspectives on the Dissertation's Progression

Through the research questions we identify what we investigate in each chapter. To understand why we arrange the chapters in the sequence we do, and how each connects to the others, we examine the dissertation's progression from two complementary perspectives: capability escalation and knowledge escalation.

1.7.1 Capability Escalation

We build from Observe through Structure, Adapt, Augment, Execute, Assess, and finally Synthesize. This sequence is not a logical arrangement in the abstract; we derive it from the structure of the E-CPSS design problem itself. We cannot Structure decisions without first Observing the state, because the decisions we must make depend on a state characterization we have not yet produced. We cannot Adapt across state transitions without first Structuring decisions within a state, because to adapt we must know which decisions connect the current state to the target. We cannot make productive use of Augmentation without a structurally sound design process to augment. A designer whose augmented cognition still operates at human speed cannot address an E-CPSS that evolves faster than they can deliberate, and so we must add the capability of Autonomous Execution. And when designers have not assessed a strategy socially, they have not met the legitimacy requirement of Strategic Engineering, however well the strategy performs against their technical measures and however autonomously they can execute it.

In Chapter 3, we build the Observe capability: we can read the state of an E-CPSS across its three spaces using a principled, design-relevant modeling method and interpret outputs as system state characterizations rather than abstract statistical patterns. In Chapter 4, we build the Structure capability: we can organize decisions within an E-CPSS state into a coupled, multi-level decision map with explicit robustness guarantees. In Chapter 5, we build the Adapt capability: we can derive adaptive intervention strategies from desired future states and update them through feedback as

the system evolves. In Chapter 6, we build the Augment capability: we can navigate high-dimensional trade-off spaces with interpretable, query able computational assistance that expands cognitive reach without removing judgment. In Chapter 7, we build the Execute capability: we can deploy autonomous agents that execute strategies and accumulate shared knowledge at the timescale of the E-CPSS. In Chapter 8, we build the Assess capability: we can evaluate the social legitimacy of a designed intervention through cognitively grounded stakeholder simulation before real-world deployment.

1.7.2 Knowledge Escalation

Through capability escalation we describe what we can do. Through knowledge escalation we describe what we know after each chapter, and how we map the knowledge we produce onto the $K(s_i)$ component of the DSIM and carry it forward into our observation of subsequent states.

In Chapter 3, we produce predictive knowledge: a structured predictive model and SHAP-attributed importance values through which we characterize the part each tri-space feature plays in the behavior we observe. In Chapter 4, we produce structural knowledge: a decision map organizing interventions by type, level, and coupling strength, along with DCI values through which we characterize the robustness of candidate solutions. In Chapter 5, we produce dynamic knowledge: the inverse design pathway from current state to target state, DCI-enforced transition conditions, and feedback-driven goal update trajectories across multiple temporal states. In Chapter 6, we produce generative knowledge: validated knowledge statements organized as a query able Knowledge Graph, capturing design heuristics derived from structured scenario exploration. In Chapter 7, we produce experiential knowledge: accumulated policy experience of autonomous agents across training episodes, organized through meta-learning architectures that allow transfer across contexts. In Chapter 8, we produce simulated knowledge: a pre-deployment map of stakeholder response patterns, resistance sources, and legitimacy conditions, organized by agent role and cognitive state.

Each type of knowledge we produce at state s_i becomes part of $I(s_{i+1})$. By the time we reach Chapter 8, we draw simultaneously on all six knowledge types when we observe the system state. In each successive state characterization we draw on the intelligence we have accumulated at all prior

states, and this cumulative growth is the learning loop of the DSIM: the primary architectural difference between what we build here and any collection of independent design methods.

1.8 Goals and Focus in Dissertation

1.8.1 Problem to be Addressed

The problem we address in this dissertation is the design of a coherent, state-indexed framework for engineering adaptive interventions in Evolving Cyber-Physical-Social Systems. To design E-CPSS across state transitions under uncertainty we must resolve several interconnected challenges simultaneously, and we develop a principled response to each one.

We develop a method for observing sociotechnical system states through which we organize heterogeneous signals from cyber, physical, and social spaces into a structured, design-relevant state characterization. We develop a method for structuring multi-level decisions under uncertainty, through which we explicitly encode the hierarchical and cross-space coupling characteristic of E-CPSS interventions. We develop a method for designing adaptive strategies across state transitions, working backward from desired future states and incorporating feedback through which designers maintain robustness as the system evolves. We develop a framework for augmenting designer cognition with generative AI, grounded in validated knowledge extraction protocols through which we make the guidance we produce both interpretable and trustworthy. We develop autonomous agent architectures for executing adaptive strategies at the timescale of the E-CPSS, bounded throughout by the governance constraints the designer specifies. We develop a pre-deployment social legitimacy assessment method grounded in the cognitive architecture of human judgment, producing predictions of stakeholder response that designers use to diagnose sources of resistance before they deploy any intervention.

Through the methods we develop across Chapters 3 through 8, we address all six challenges, and in Chapter 9 we demonstrate that these methods, understood through the DSIM framework, constitute a coherent architecture for Strategic Engineering of E-CPSS.

1.8.2 Contributions Through this Work

We organize the contributions of this dissertation in two tiers, grounding them in Boyer's taxonomy of scholarship. At the level of method, we contribute through the Scholarship of Discovery: across six research chapters, we develop new methods, new mathematics, and new architectures, and through each we produce knowledge that did not previously exist in this form. At the level of synthesis, we contribute through the Scholarship of Integration: we demonstrate in Chapter 9 that these six method-level contributions, taken together, constitute a coherent and sufficient framework for the strategic engineering of E-CPSS, and that through this framework we collectively address all eleven design requirements that we identify in Chapter 2. We present the contributions in the order in which we develop them, describing what each contribution is, what we demonstrate through it, and what new knowledge we produce from it.

Contribution 1: CPSS-Structured Predictive Modeling for State Observation. In Chapter 3, we develop a method for organizing observational features according to the tri-space structure of an E-CPSS before model construction, building a random forest regression model on this structured representation, and interpreting outputs through SHAP-based feature attribution mapped back to the cyber, physical, and social spaces. Through this method, we construct I(si), the structured tri-space state characterization that forms the Observation Layer of the DSIM. Applied to the ALAN and bird strikes test problem, we demonstrate that through this approach we construct a predictive model that is both accurate and interpretable for design decisions and we identify light pollution level as the strongest predictor. This constitutes new knowledge in the form of a principled, design-oriented method for constructing tri-space state characterizations of E-CPSS, with demonstrated implications for public policy and aviation safety.

Contribution 2: Integrated Policy Designing Decision Model for Robust Multi-Level Decision Structuring. In Chapter 4, we develop the Integrated Policy Designing Decision Model, a coupled compromise Decision Support Problem formulation that encodes decision types, interaction strengths, and hierarchical levels across macro, meso, and micro scales of an E-CPSS intervention, with Design Capability Index constraints enforcing robustness under stakeholder priority uncertainty. Applying the method to the Agriculture 5.0 test problem, we demonstrate that through this method we produce satisficing solutions that remain robust across stakeholder priority configurations. Through this contribution we produce $D(s_i)$, the intra-state decision map of the

DSIM, and we contribute new knowledge in the form of a coupled decision method for designing robust multi-level satisficing strategies in E-CPSS under uncertainty.

Contribution 3: Goal-Oriented Inverse Design Method for Adaptive Strategy Across State Transitions. In Chapter 5, we develop the Goal-Oriented Inverse Design Method, through which we derive adaptive intervention strategies by working backward from desired future states, enforce DCI constraints at every state transition, and update goals through feedback when observed performance deviates from projections. Applied to the UAE sustainable development test problem across three temporal states (2017, 2023, and 2030), we demonstrate that through this method we produce adaptive strategies with explicitly characterized robustness properties, and that by updating goals through feedback we preserve the viability of the strategy across a multi-decadal planning horizon. Through this contribution, we produce, D(si to si+1), the inter-state decision map of the DSIM, and we contribute new knowledge in the form of an inverse design framework for deriving adaptive strategies across E-CPSS state transitions under uncertainty.

Contribution 4: Generative AI and Knowledge Graph Framework for Scalable Trade-Off Navigation. In Chapter 6, we develop a three-stage framework through which a large language model extracts candidate knowledge statements from structured scenario data, each statement is validated quantitatively against scenario outcomes before ingestion into a Knowledge Graph, and the validated graph supports interactive querying and adaptive infeasibility recovery through a conversational interface. Applied to the hot rod rolling steel manufacturing test problem across 169 scenarios, six objectives, and two coupled design levels, we demonstrate that through this framework a designer can navigate a high-dimensional satisficing space with interpretable and actionable guidance. Through this contribution we produce the Augmentation Layer of the DSIM and we contribute new knowledge in the form of a generative AI and Knowledge Graph framework for supporting design decision-making in high-dimensional E-CPSS.

Contribution 5: A Knowledge-Governed Autonomous Agent Architecture for Real-Time Adaptive Execution. In Chapter 7, we develop an architecture for autonomous strategic execution in E-CPSS. Through it, we compile the agent's aspiration from a Knowledge Graph that we update as facts arrive, we hold the value-learning channel apart from the aspiration channel, and we constrain the agent's action space throughout training so that we bound every policy within the governance limits a designer specifies. Applied to the supply chain resilience test problem, we

demonstrate that through this architecture we achieve the performance characteristics we predict in the hypothesis. This contribution constitutes the Autonomous Execution Layer of the DSIM, and it stands as new knowledge in the form of a principled architecture for autonomous, knowledge-governed strategic execution in multi-agent E-CPSS under non-stationary conditions.

Contribution 6: Cognitively Grounded Stakeholder Agent Architecture for Pre-Deployment Social Legitimacy Assessment. In Chapter 8, we develop a memory-emotion coupled stakeholder agent architecture in which we treat arousal as a proxy for epistemic uncertainty and scale the influence of an agent's emotional state on memory retrieval accordingly, grounded in predictive-processing principles, and in which all agents operate within a shared neurosymbolic knowledge graph with role constraints enforced through a rule system. Applied to the carbon emission taxation test problem, we demonstrate that the cognitively grounded agents produce behavioral outputs measurably distinct from rational-actor baselines and from them we obtain diagnostic pre-deployment intelligence about stakeholder resistance and about the conditions under which stakeholders are likely to grant or withhold social legitimacy. This contribution produces the Social Assessment Layer of the DSIM and constitutes new knowledge in the form of a cognitively grounded, neurosymbolically anchored stakeholder simulation architecture for pre-deployment social legitimacy assessment.

Synthesis Contribution: The Decision-State Information Model. The synthesis contribution of this dissertation is the Decision-State Information Model itself, formally defined as the six-tuple {S, I, D, M, K, T}. The six method-level contributions each correspond to a specific DSIM component and operational layer. The knowledge we produce at each state we carry forward into our observation of the next through the $K(s_i)$ to $I(s_{i+1})$ connection, and through the transition condition $T(s_i \text{ to } s_{i+1})$ we ensure that we have satisfied robustness, knowledge retention, execution effectiveness, and social legitimacy before we authorize any state advance. In Chapter 9, we demonstrate that through the DSIM, as we constitute it from these six contributions, we collectively operationalize all eleven E-CPSS design requirements that we establish in Chapter 2 and that this completeness holds across the five different domains in which we validate the individual contributions. This is a contribution at the level of Boyer's Scholarship of Integration: we show that the six discoveries have a structure together that none of them has individually, and

that this structure constitutes a principled architecture for the strategic engineering of Evolving Cyber-Physical-Social Systems.

1.8.3 Way Forward

We go forward on the escalation logic we have established. In Chapter 2, we prepare the theoretical and conceptual ground by anchoring the fundamental of E-CPSS, tracing the intellectual lineage of the design object from Cyber-Physical Systems through Cyber-Physical-Social Systems to E-CPSS, developing the tri-space coevolution model, establishing the eleven design requirements, and mapping the seven research gaps to the research questions. In Chapters 3 through 8, we develop the six method-level contributions in sequence, and in each we address one escalating challenge and construct one operational layer of the DSIM. In Chapter 9, we assemble the complete argument, map the DSIM against all eleven E-CPSS design requirements, evaluate all hypotheses, and answer the Principal Research Question.

1.9 Validation Strategy and the V&V Square

hen we propose a new framework for engineering design, we must validate it against criteria we state in advance. We organize the validation strategy using the Verification and Validation Square, which we adapt from Seepersad and co-authors, and through which we partition the validation of a design method into four quadrants along two axes: theoretical versus empirical and structural versus performance validity.

Through Theoretical Structural Validity we ask whether the framework and its components are logically sound, whether the research questions are correctly formulated, and whether the hypotheses are well-grounded in a defensible argument for why they should hold. We establish Theoretical Structural Validity in this chapter through the motivation and problem identification argument, in Chapter 2 through the critical literature review and research gap mapping, and in the Frame of Reference section of each research chapter, where we argue why we judge the test problem suitable, by criteria we state there, and why we treat the method we develop as the right response to the design challenge.

Through Theoretical Performance Validity we ask whether a designer can use the framework and its contributions beyond the specific test problems on which we demonstrate them. We address Theoretical Performance Validity in Chapter 9 through the generalizability argument, in which we map the DSIM against all eleven E-CPSS design requirements and demonstrate that the completeness of the framework is not an artifact of the domains we choose for validation.

Through Empirical Structural Validity we ask whether the test problems we choose exercise the parts of each method we claim to validate. We establish Empirical Structural Validity in each research chapter through explicit justification of the test problem selection.

Through Empirical Performance Validity we ask whether we obtain the outcomes we predict in the hypothesis when we apply the method to the test problem. We address Empirical Performance Validity in each research chapter through the results sections, where we present quantitative outcomes and compare them against the predictions of the hypothesis.

In Figure 1.3, we present the Verification and Validation Square and annotate each quadrant with the specific dissertation content through which we address it.

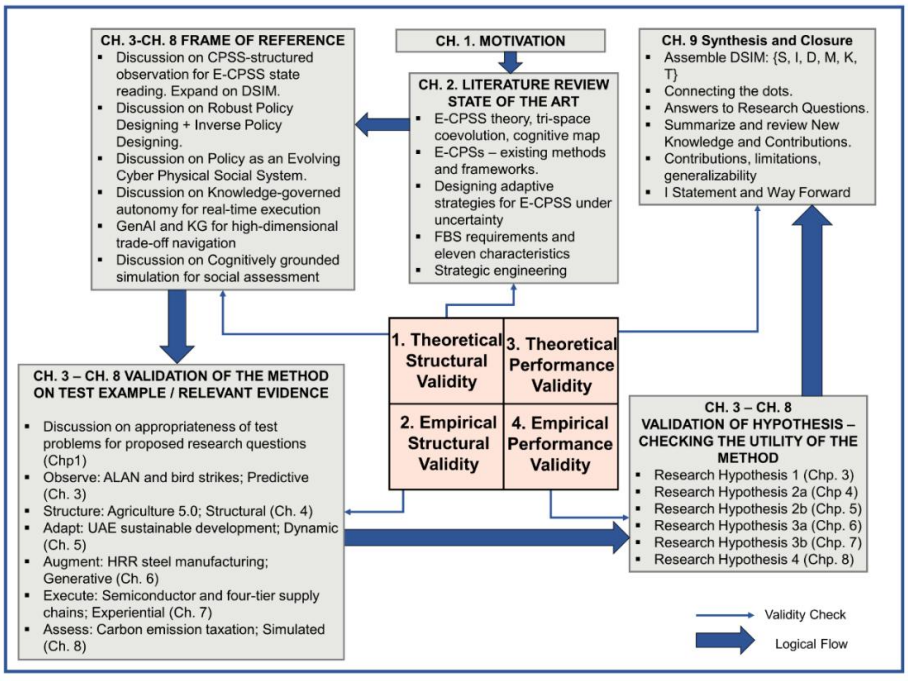


Figure 1.3: Verification and Validation Square

We provide theoretical arguments for the proposed framework and evaluate individual components through separate test problems. We do not evaluate the complete DSIM through a single E-CPSS in which we pass information and decisions from Layer 1 through Layer 5. We identify that end-to-end evaluation in Chapter 9 as the direction for future work we consider most pressing, and we state here what we have and have not evaluated so that a reader can judge the scope of our contribution.

1.10 Organization of the Dissertation

We have organized the nine chapters in two parts. In Part I, comprising Chapters 1 through 8, we establish the foundation, develop the theoretical framework, and present the six method-level contributions. In Part II, comprising Chapter 9, we synthesize the contributions into the DSIM, answer the research questions, and present directions for future work.

In this chapter, we establish the motivation and problem identification. In Chapter 2, we develop the critical literature review and justification of the research questions, anchoring the dissertation in the foundational E-CPSS and mapping seven research gaps to the research questions. In Chapter 2, we provide the theoretical grounding on which we build every subsequent Frame of Reference section, and from which we trace every research question to a named gap in the body of knowledge..

In Chapter 3, we address RQ1 and construct DSIM Layer 1, the Observation Layer. Using the ALAN and bird strikes test problem, we develop the CPSS-structured predictive modeling method and demonstrate that through it we produce both higher predictive accuracy and more design-interpretable feature attribution than a designer achieves through undifferentiated modeling approaches.

In Chapter 4, we address RQ2a and construct the intra-state component of DSIM Layer 2, the Decision Structuring Layer. Using the Agriculture 5.0 test problem, we develop the Integrated Policy Designing Decision Model and demonstrate that through it we produce satisficing solutions that remain robust across stakeholder priority configurations.

In Chapter 5, we address RQ2b and complete DSIM Layer 2. Using the UAE sustainable development test problem, we develop the Goal-Oriented Inverse Design Method and demonstrate

that through it we produce adaptive strategies with explicitly characterized robustness properties that a designer cannot achieve through a conventional forward design approach.

In Chapter 6, we address RQ3a and construct DSIM Layer 3, the Augmentation Layer. Using the hot rod rolling steel manufacturing test problem, we develop the generative AI and Knowledge Graph framework and demonstrate that through it we can navigate a 169-scenario, six-objective, two-level design space with interpretable and actionable design guidance.

In Chapter 7, we address RQ3b and construct DSIM Layer 4, the Autonomous Execution Layer. Using the supply chain resilience test problem, we develop an autonomous agent architecture and demonstrate that through it we achieve the performance characteristics we specify in the hypothesis.

In Chapter 8, we address RQ4 and construct DSIM Layer 5, the Social Assessment Layer. Using the carbon emission taxation test problem, we develop the memory-emotion coupled stakeholder agent architecture and demonstrate that through it we produce behavioral simulations measurably distinct from rational-actor baselines, yielding diagnostically useful pre-deployment intelligence about stakeholder resistance and legitimacy conditions.

In Chapter 9, we synthesize all six contributions into the DSIM, map the complete framework against the eleven E-CPSS design requirements, evaluate all hypotheses against the accumulated empirical evidence, answer the Principal Research Question, and present the generalizability argument and directions for future research.

In Figure 1.4, we present the Dissertation Roadmap, in which we show the nine-chapter flow, annotate the escalation arc and the DSIM layer contributions at each chapter node, and record the knowledge types we produce and carry forward and the V&V quadrants we address at each stage.

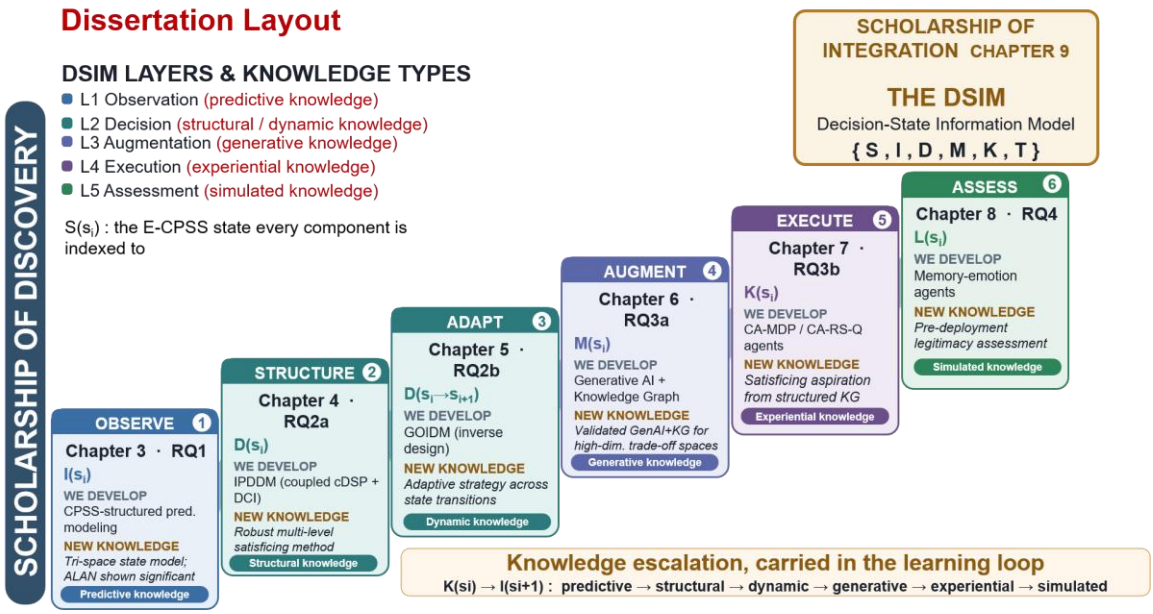


Figure 1.4: Dissertation Roadmap

1.11 What We Have Presented in Chapter 1 and What Comes Next

In this chapter, we have established the motivational and structural foundation of the dissertation. We opened with the argument that designers increasingly face engineering problems whose cyber, physical, and social spaces change one another in ways that classical engineering design does not accommodate, and we grounded that claim in the theoretical foundation we develop in Chapter 2, in Allen and co-authors [2] on E-CPSS design, in Cardin [4] on designing for temporal uncertainty, and in Rittel and Webber [1] on the limits of systems-engineering approaches to open sociotechnical problems. We established Strategic Engineering as the paradigmatic commitment through which we orient our methods toward maintaining value across uncertain, evolving futures, drawing on Herbert Simon's [6] satisficing principle as the decision orientation we adopt for high-dimensional, incompletely specified problems.

We identified information asymmetry and intelligence asymmetry as the two structural sources of difficulty in every E-CPSS design challenge and argued that we must resolve both simultaneously rather than treat them as independent problems. We presented six escalating challenges, Observe, Structure, Adapt, Augment, Execute, and Assess, as the specific manifestations of these asymmetries across the dissertation's five domains and showed that for each we require a distinct

method, and for all six together a unified architecture. We introduced the Decision-State Information Model, formally defined as the six-tuple {S, I, D, M, K, T}, as the synthesis target: the architecture through which we establish, at every state of an E-CPSS, what information is available to us, what decisions we must make, which methods we can apply, what knowledge we produce, and what must be true before we advance the system legitimately.

We stated the Principal Research Question and four secondary research questions with their associated hypotheses and expected new knowledge, grounded the derivation of each question in the fundamental assumptions we adopt, and explained the sub-question structure of RQ2 and RQ3 in terms of the DSIM layers we address through them. We enumerated the contributions at both the discovery and integration levels, and organized the validation strategy around the Verification and Validation Square.

We ground every method we develop in Chapters 3 through 8 in the theoretical framework for E-CPSS we describe in Chapter 2. Through each research question we address a specific capability that designers need, and we develop and validate that capability in the corresponding chapter. We anchor every hypothesis in design requirements we derive from the E-CPSS paradigm before we operationalize them as research contributions. We move from motivation to method by way of theory. In Chapter 2, we carry out that preparatory work: we trace the intellectual lineage of the E-CPSS design object, develop the tri-space coevolution model, establish the eleven design requirements, map the seven research gaps to the research questions, and provide the theoretical justification for why each research question is warranted given the state of the field. That is where we go next.

Theoretical Structural Validity of PRQ and RQ1 through RQ4 in Chapter 1

In this chapter, we justify the necessity of investigating the problem of designing adaptive interventions for E-CPSS under uncertainty. We establish Strategic Engineering as the paradigmatic frame, identify information asymmetry and intelligence asymmetry as the two root sources of difficulty, present the PRQ, RQ1, RQ2a, RQ2b, RQ3a, RQ3b, and RQ4 with their hypotheses and expected new knowledge, and describe the validation strategy we follow throughout the dissertation. Through the argument we construct in this chapter we establish Theoretical Structural Validity for all research questions and for the Principal Research Question.

References

[1] Rittel, H. W. J., & Webber, M. M. (1973). Dilemmas in a general theory of planning. Policy Sciences, 4(2), 155-169. https://doi.org/10.1007/BF01405730

[2] Allen, J. K., Nellippallil, A. B., Ming, Z., Milisavljevic-Syed, J., & Mistree, F. (2023). Designing evolving cyber-physical-social systems: Computational research opportunities. ASME Journal of Computing and Information Science in Engineering, 23(6), 060815. https://doi.org/10.1115/1.4062883

[3] Sobb, T., Turnbull, B., & Moustafa, N. (2023). A holistic review of cyber-physical-social systems: New directions and opportunities. Sensors, 23(17), 7391. https://doi.org/10.3390/s23177391

[4] Cardin, M.-A. (2014). Enabling flexibility in engineering systems: A taxonomy of procedures and a design framework. ASME Journal of Mechanical Design, 136(1). https://doi.org/10.1115/1.4025704

[5] Jiao, R., Commuri, S., Panchal, J., Milisavljevic-Syed, J., Allen, J. K., Mistree, F., & Schaefer, D. (2021). Design engineering in the age of Industry 4.0. ASME Journal of Mechanical Design, 143(7), 070801. https://doi.org/10.1115/1.4051041

[6] Simon, H. A. (1955). A behavioral model of rational choice. The Quarterly Journal of Economics, 69(1), 99-118. https://doi.org/10.2307/1884852

CHAPTER 2

CRITICAL LITERATURE REVIEW AND JUSTIFICATION OF RESEARCH QUESTIONS

Background and Context

In Chapter 1, we lay out the intellectual commitments of this dissertation: Strategic Engineering as the paradigm, the E-CPSS as the design object, two structural asymmetries as the source of difficulty, and the Decision-State Information Model as the target we are building toward. We do more than state these commitments. We ground them in prior scholarship and explicit design requirements. Before the first line of method can be written, we need to know precisely what an E-CPSS requires of the designer and why those requirements are not yet met. That is the work we do in this chapter.

In this dissertation, we respond to a documented gap in how researchers design evolving sociotechnical systems. We do not select the DSIM structure arbitrarily. We require that through each DSIM layer we operationalize a named design requirement, and we derive each research question from a documented gap in current scholarship and practice. In this chapter, we establish the theoretical basis for the six research chapters that follow. We draw on prior E-CPSS scholarship to explain the tri space coevolution model, classify the design requirements through Function, Behavior, and Structure, and identify eleven characteristics that designers need to consider when they design an E-CPSS.

We trace the intellectual lineage of the design object from Cyber-Physical Systems through Cyber-Physical-Social Systems to E-CPSS, establishing along the way why adding the social space as a first-class design dimension fundamentally changes what designing means rather than simply adding a new variable to an otherwise familiar problem. We present the tri-space coevolution model and the cognitive map of E-CPSS evolution, develop the FBS requirements framework, and present the eleven E-CPSS characteristics. We close by mapping seven research gaps to the

Principal Research Question and to each of our four secondary research questions, so that every question in this dissertation has a named and justified origin before a single method is proposed.

We limit the review in this chapter to the theoretical foundations of E-CPSS design. In the Frames of Reference in Chapters 3 through 8, we review scholarship that relates directly to each method. What we establish here supports every research question and confirms the eleven requirements we operationalize through the DSIM layers. In every subsequent chapter we build on this foundation.

Preface

In moving from the Cyber-Physical System to the Cyber-Physical-Social System (CPSS), researchers incorporate the social space as an independent entity that interacts with the cyber and physical spaces, and with it human cognition, coordination, and other behavioral factors. We recognize that the social space continuously evolves in a dynamic social environment. Researchers have only recently begun to develop systematic design guidance through which a designer unifies evolvability with an explicit social cognition dimension. The idea of Evolving CPSS (E-CPSS) is relatively unexplored. Designers build an E-CPSS so that it adjusts its behavior and reorganizes its components as stakeholder requirements and environmental conditions change. By exploring the E-CPSS paradigm, we introduce the concept of continuous adaptation, through which we account for the influence of time and dynamic social contexts on decision-making. We establish a foundation for E-CPSS and outline the requirements for designing an E-CPSS from functional, behavioral, and structural perspectives. In this chapter we highlight the evolving nature of the CPSS and the pivotal role of the social entity in decision support. We have examined the key features and attributes of E-CPSSs to showcase their utility in decision-based design. Although many researchers acknowledge cognition and user interaction in their studies of CPS, few of them treat cognition, motivation, emotion, and social norms computationally as evolving state variables. We address this gap with a conceptual map. We use this cognitive map to illustrate identified attributes and aid designers in understanding the system's evolutionary mechanism, as well as identifying potential avenues of exploration and research gaps in the field.

Key Words: Cyber-Physical-Social System (CPSS), Decision Support, Design, Evolution, Human Reliability Assessment (HRA), Self-Organizability, Self-Adaptability, Social Environment

GLOSSARY

Cognition: The ability of humans or system agents to acquire, interpret, and utilize information to make decisions, influenced by evolving preferences, goals, and social contexts

Collective decision-making: The process of making decisions collaboratively and often distributed among multiple entities, subsystems, or stakeholders within the system.

Cyber-Physical-Social Space: An Integrated environment where cyber, physical, and social entities interact.

Decision Support: Leveraging methods to aid the system decision-makers, whether they are human or internal to the system, to enhance system performance, adapt to changes, and manage risks throughout the lifecycle.

Endogenous social space: Through this component we capture the internal social dynamics — stakeholder interactions, behavioral responses, and trust formation — that emerge from within the system.

Exogenous social space: Through this component we represent external social influences such as cultural norms, societal expectations, and policies that lie outside the system's control.

Interactions refer to the dynamic relationships and exchanges, such as communication, coordination, and influence, that occur among various system components.

Scenario recognition: The process of identifying and understanding different scenarios or situations by analyzing data from various sources to identify patterns and anomalies indicating the occurrence of specific scenarios

Self-Adaptability: The capability of a system to adjust or modify its behavior, structure, or function autonomously in response to changing conditions or requirements.

Self-organizability: The capability of the system to autonomously and collaboratively organize its components and resources to achieve the desired system behavior or function.

Stakeholders: Individuals, groups, or entities, including other CPSSs, that have an interest or concern in the success and outcomes of the CPSS

System Behavior: Dynamic and observable aspects of how an E-CPSS operates and responds to various stimuli or inputs over time.

System Structure: The architecture, composition, and arrangement of the components that make up the system

2.1 Frame of Reference

During the last decade, researchers have debated the role of the social element in engineered systems. The profound impact of human behavior and cognition on various systems, including manufacturing, automotive, and social systems, is evident [1] [2]. Researchers associated with Industry 4.0 derived significant utility from it during the last decade, ushering in Industry 5.0, which scholars describe as ‘a vision of an industry that aims beyond efficiency and productivity as the sole goal, and reinforces the role and contribution of industry to society. Through Industry 5.0, policymakers and firms shift their focus from shareholders to stakeholders and align their objectives with people's values and preferences [3]. In doing so they adopt a human-centric approach rather than consider business perspectives alone.. This shift allows the fifth Industrial Revolution to adopt a human-centric approach, transcending the sole consideration of business perspectives. The concept of CPS, originally proposed to address the tight integration of computation with physical processes [4], laid the foundation for subsequent evolutions that integrate social and cognitive dimensions. Researchers note significant changes in design practice as CPS have evolved into CPSSs that explicitly incorporate human behavior, cognition, and broader social and cultural contexts. While self-evolving CPS research is increasingly mature, systematic design guidance that unifies evolvability with an explicit social-cognition dimension is still nascent. We term these systems' E-CPSSs[1]. In addition, while self-adaptive and self-evolving CPS have been increasingly discussed in recent literature [5] [6], researchers have not yet integrated these concepts with the socially embedded decision-making at the center of CPSS. In this chapter, we refer to the term E-CPSS to emphasize this socio-technical convergence and its implications for design. Our use of E-CPSS does not imply a rigid classification of CPSS into evolving or non-evolving types. Rather, it reflects a design orientation: the intentional structuring

[1] The term does not imply that other CPSS lack adaptive capacity; it designates systems whose evolvability explicitly accounts for social-space cognition and trust.

of CPSS to support evolution across both technical and social dimensions. We argue that cyber-physical innovation depends on treating the social dimension as a time-varying design entity. We therefore posit that designers must conceive the next generation of complex engineered systems as Decision-Centric, Socially Cognizant, Continuously Evolving Cyber-Physical-Social Systems. Designers require a unifying metamodel that couples three coevolution loops (physical, cyber, and social) through an explicit, measurable cognition layer. We argue that through a framework coupling the three spaces to an explicit cognition layer, designers can verify self-adaptation, keep autonomy auditable, and earn stakeholder legitimacy.

2.1.1 Integrating Human Factors into the CPSS

A CPS is designed to combine computation, sensing, communication within cyber entities, and actuation of physical systems to accomplish time-sensitive functions with varying levels of interaction between the cyber and physical entities and their associated environment [7, 8]. A CPSS differs from A CPS by introducing the social space as a functionally distinct dimension that captures the influence of human behavior, collective dynamics, and societal context. Through it we represent the emergent interactions, shared norms, and cultural constructs that shape and are shaped by the system, making it essential to model social entities explicitly rather than treat them as a derivative of other spaces. Through the CPSS we gain an important perspective on the evolving relationships and interactions among humans, computers, and physical environments in the context of human behavior. In CPSS, the social system is an independent entity that can interact with the cyber- and physical worlds, where humans act as resources (sensors in providing information, services, knowledge, etc.) or consumers (end-users of information, services, and knowledge). [9]. Horvath suggests that cognition be involved, but through the intellectualization of CPSS [10]. However, as mentioned above, there is more to consider in human factors than cognition.

2.1.2 Addressing the Entities: Cyber, Physical, and Social Spaces

The CPSS is the integration of cyber, physical, and social spaces, Figure 2.1. Cyberspace is a metaphorical space that represents a conceptual layer enabling communication, computation, and coordination of information. In cyberspace, seamless interaction and data sharing among social agents occur, incorporating human features like judgment, experience, cognition, decision-making,

and behavior, making it evident that introducing a social space adds an untapped dimension of integrative mental space comprising human knowledge, sociocultural elements, and mental capabilities, as well as the ability to make decisions [11]. Physical space is the material and energetic reality in which all action occurs; it may be sensed, actuated, or engaged, but it is confined to these physical actions rather than serving as an information exchange medium. Physical space is anchored in the actuation medium and a source of data-producing entities through tangible entities that exist in the physical world, such as sensors, robots, infrastructure, and devices. Social space is the metaphorical layer through which individual entities or groups interact with malleable social structures to represent, express, and react to needs, wants, and values without explicitly exchanging information or engaging in physical interaction with other social agents. Stakeholders express preferences, and designers or the agents they program act on those preferences in the physical space. These social systems are embedded and interact with other social systems, designed to behave socially [10].

A CPS is well-defined in academic literature as a complex system that integrates information gathering, computation, actuation, and responsive feedback mechanisms, often in the context of networks of devices, control architectures, and distribution management, among others. The CPSS is an extension of CPS, including the social space [12] [7, 13, 14] [15] [16] [17]. For a CPSS to exist, each space must interact with all other spaces: Physical-Social, Physical-Cyber, and Cyber-Social. Through these interactions designers define what a CPSS does, and they analyze the relationships among the three spaces so that the system delivers the services stakeholders ask for [18]. As an example of a CPSS, consider a package delivery system. The physical space includes the packages and delivery routes; while the cyber space contains location, directory, and information about which packages are assigned to each route. The social/cognitive space includes preferences of customers about their package deliveries and some decision-making capability about the timing and sequence of deliveries. This approach mirrors human decision-making in situations where conflicting and contradictory objectives exist regarding varied goals, preferences, opinions, and beliefs, and their impact on the underlying problem structures.

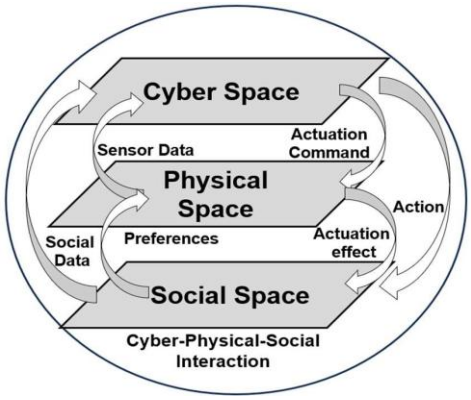


Figure 2.1: Cyber-Physical-Social Spaces and Interactions

2.1.3 E-CPSS Foundations: Objectives and Organization

System designers must consider the evolution and integration of social space in complex systems such as CPSSs, the evolving nature of the systems anchored in the time-varying environment. Researchers attribute system evolution to several factors, and they observe change in technology, function, architecture, cognition, social relations, and environment. Designers must consider these forms of self-evolution together at the design stage. We refer to these instances at any particular period as 'states' [19]. Designers may define any number of states in an evolutionary cycle, and they select the states according to what they intend to explore. Every system changes over time to adapt itself to the dynamic environment. In this chapter, we focus on addressing the relatively underexplored paradigm of E-CPSS. With the advent of real-time decision-making, designers are considering the operational states of the developed system during the conceptualization and design stages. It is still debated whether designing for self-evolution constitutes a conceptual or a detailed design task.

Designers must attend to evolution, because they must build systems that respond as people change the social environment. Designers must speculate on and inform a system about how to respond to changes in external stimuli. As designers build more complex intelligent systems, they must give those systems access to context. As a self-learning system gathers context data, it may intelligently reassess its behavior in response to a constantly changing environment with minimal human intervention [20].

Other authors analyze how systems change over time. For example, Weyns and co-authors study self-evolving computer vision with a focus on architecture rewrite loops, but they do not consider

the social-cognition aspects of the system [5]. Horvath studies Next-Generation CPS with a focus on model-based runtime evolution with a user in the loop. ADACOR [6], by Barbosa and co-authors, study with operators of manufacturing self-organizing systems [21]. Mandegari and co-authors investigate self-adaptive systems from a mathematical robust satisficing perspective, employing both backward and forward modeling approaches [A].Bhalerao and co-authors propose a design method for designing policies as E-CPSS by examining various decision couplings and levels of abstraction for a policy designing problem[B].Zhou and co-authors explore Force Field E-CPSS by reconfiguring the force vector based on the concept of trust [8]. Parallel world CPSS is based on the idea of population evolution, but is limited to the idea of sentiment mining [22]. Bucchiarone and co-authors develop collective adaptive systems which incorporate social interplay but do not include a model of cognition [23].

In this chapter, we establish the foundational principles of E-CPSS design by emphasizing the social space, where stakeholders hold individual preferences, objectives, and goals. The interests and objectives of stakeholders are constantly changing with the progression of time (states), influenced by the external environment. The problem here is threefold: (a) a CPSS will evolve, (b) a social entity will evolve and induce change in the CPSS, and (c) interests in social space constantly change with states. Through this paradigm designers create effective systems in manufacturing, automotive, and other industries, and address grand challenges such as environmental justice, climate adaptation, equitable resource allocation, food insecurity, and healthcare.

The foundational knowledge concerning E-CPSS must be more frequently reported in the state of the art. It is imperative to conduct a systematic study to explore, identify, and analyze the characteristics/ attributes of an E-CPSS. Furthermore, it is critical to identify promising frameworks, methods, and approaches that can be adapted to E-CPSS, thereby further establishing computational capabilities. In this position chapter, we establish a list of requirements to guide system designers in effectively addressing the complexities and demands of E-CPSS. We write this chapter to synthesize, critique, and extend scholarly foundations rather than to document current design cases. We emphasize that functional complexification in the cyber and physical spaces triggers new patterns of social interaction. Hence, the progression for CPS to CPSS and the shift from static to evolving behavior should be viewed as mutually reinforcing trends.

2.1.4 Paradigm-Level Evolution by means of the Knowledge Spaces

In this section, we provide an explicit clarification of how we differentiate between the concepts of adaptation and evolution within the context of Cyber-Physical-Social Systems (CPSS). This distinction is essential for understanding the layered temporal and functional nature of change in E-CPSSs, especially since, in many parts of this chapter, including the cognitive map and the decision-making framework, we build upon this conceptual hierarchy. Historically, evolution is defined inconsistently in various subfields of systems, and these definitions often deviate from the biologically inspired sources. In this chapter, we define evolution as a change in the current context and paradigm of a system at discrete time intervals of variable length due to either changes in the external environment of the system or internal emergent properties arising due to interactions among components, such that it is no longer useful to consider the system in the previous context or paradigm. This definition diverges from older models of system evolution, such as the Deming cycle (plan-do-study-act) [24], which underpins the iterative enhancement models of evolutionary delivery as a response to the limitations of waterfall development models in software engineering [24]. Through the older definitions researchers punctuate evolutionary stages by deliverables [19]. We find this less useful for social systems, in which groups and priorities reorganize gradually, in a manner closer to traditional Darwinian accounts [25]. Like biological evolution, E-CPSSs may be described as a system that evolves to accept adaptations that suffice the requirements of the latest paradigm and discarding maladaptive features. A parallel has been reported for physiological control systems defining evolution as a "semicoordinated, multidimensional, non-homeostatic controlling system with operators functioning on 'strange' coordinates and defined as much by their nonrules as by their rules while providing a piecewise control on both overdetermined and underdetermined subfunctions in a variety of partly overlapping and partly discontinuous regions where, using simple approximations, it computes statistically 'good enough' responses from incomplete or even 'inappropriate' data" [26]. While targets may be set by a designer, control is ceded to the control system for the serendipitous emergence of properties between generations. Given equal opportunity for all possible adaptive traits, the designer may only act upon these evolutionary stages, inasmuch as an agent of artificial selection through culling of undesirable adaptations to the new paradigm. The social system in an E-CPSS may never be as simple as a human agent interacting with a controller.

With these constraints in mind, some statistical ramifications must be considered. It is not appropriate to describe an E-CPSS using uniform time intervals, such as yearly or quarterly assessments. Instead, evolutionary stages become apparent through continuous observation, allowing for the detection of time intervals. While it is astronomically convenient to assign precise dates for the start of a season based on the solstice, in practice, summer heat will last much longer in one location than in another. As in assessments of evolutionary systems described by Deming and subsequent researchers, a designer must choose metrics that designate and describe changes to the system environment, which can be tested continuously for evidence of change. Consider, for example, using a thermometer to determine when summer ends rather than relying on a calendar. With changes identified, descriptive statistics become more useful for studying the state of intergenerational evolution rather than time-domain statistics. Exceptions apply; daily temperature at solar noon may still be a useful metric within the description of a season.

We clarify the relationship between adaptation and evolution, and explain that continuous self-adaptation contributes to, but does not guarantee, system evolution. Adaptation refers to localized, context-specific changes in a system's behavior, structure, or policy that enable the system to maintain or improve its performance in response to immediate internal or external stimuli. For example, a system may adjust resource allocation in response to demand surges or shift communication bandwidth when congestion is detected. These changes are operational and reversible, designed to preserve functional stability without altering the overall system architecture or goals.

Evolution, in contrast, denotes a longer-term transformation of the system's architecture, function, or behavioral patterns. Evolution occurs when adaptations accumulate and fundamentally alter how the system operates or what it is intended to do. It may involve changes in stakeholder roles, system functions, or even design values. Unlike adaptation, evolution is a strategic and often irreversible process.

To capture this distinction, we now introduce a taxonomy of system-level and paradigm-level evolution, with particular emphasis on the role of knowledge spaces as engines of paradigm change.

System-Level Evolution

System-level evolution occurs when designers or operators implement persistent changes in system behavior and structure across a series of states t1, t2, …, tn. Unlike adaptation, which is often local and transient, these changes accumulate into new stable patterns that redefine the system's identity or purpose. For example, a transportation coordination platform might initially evolve by integrating new scheduling algorithms, then by adding demand-responsive routing, and later by embedding predictive maintenance functions. Each step is an accumulation of adaptations, and when enough of these are retained, the system has effectively evolved into something qualitatively different from its starting form.

In short, system-level evolution describes how this system changes over time: through the retention and consolidation of adaptations, micro- and meso-level shifts eventually cross thresholds and become macro patterns that persist.

Paradigm-level evolution

Researchers introduce new knowledge spaces, and through them designers gain functions that change both what they can design and how they and their reviewers validate a design.

For instance, when a designer incorporates a trust knowledge space, the system adjusts its level of autonomy against the expectations the designer encodes, and the designer reallocates authority as reliability changes, so that operators do not over-rely on it [27] [28]. Introducing a cognition knowledge space supports functionalities like collective sensemaking and adaptation to stakeholder feedback, which in turn shift evaluation criteria toward legitimacy, transparency, and resilience [28] [29]. Similarly, embedding autonomic knowledge spaces inspired by MAPE-K (monitor–analyze–plan–execute with knowledge) provides self-management capabilities at runtime, changing the paradigm of control from prescriptive to reflexive [30].

By creating and applying these knowledge spaces, designers expand what systems can do and also reshape what counts as valid design. A system once judged solely by efficiency may now be evaluated by whether its decisions are explainable, legitimate, and socially acceptable. Thus, through paradigm-level evolution researchers redefine the "rules of the game" for an entire class of systems.

2.1.5 Indicators and Metrics for Adaptation and Evolution

Through the distinction between adaptation and evolution, we help us understand how systems change over time; however, for design practice, this distinction must be emphasized. To design an evolving cyber-physical-social system, designers and stakeholders need indicators by which they judge whether the system is adapting effectively, whether they retain those adaptations as part of longer-term evolution, and whether such changes remain aligned with their goals of resilience, legitimacy, and human-centricity. Without such indicators, adaptation and evolution risk remain abstract metaphors rather than operational processes.

Adaptation can be understood in terms of responsiveness, stability, and alignment. Responsiveness is how quickly the system adjusts as conditions change. Through stability we express whether the adjustments converge into sustainable behavior rather than oscillate or undermine prior functions. We ask whether the adaptation aligns with purpose across the cyber, physical, and social spaces.

To recognize evolution we need indicators across longer timescales. A central measure is the retention of adaptations, which refers to the proportion of adjustments that persist across states and become part of the system's enduring structure. In addition, when stakeholders adopt new evaluation criteria, shifting from pure efficiency to legitimacy or explainability, we recognize that the system is undergoing a deeper transformation. Trust calibration error, defined as the gap between human expectations of performance and actual performance, offers another useful indicator of whether evolution remains aligned with societal acceptance [27] [28]. Equally, through resilience indices we track the ability of a system to sustain core functions under disruption and validate whether evolution leads to robust and trustworthy outcomes [31] [32].

Designing Adaptation and Evolution at the Development Stage

During the development stage, the role of indicators is prospective. The aim is not to measure outcomes in the field but to anticipate possible contexts and chart out adaptation pathways. Designers model various scenarios, simulate stressors, and track how different adaptations may accumulate over time, leading to evolution. Designers judge their own preparedness here by asking: have we explored diverse scenarios, have we stated how micro-level adaptations might consolidate into macro-level change, and have we built in a way to record state histories?

Cognitive maps, discussed in Section 2.4, are especially valuable at this stage. Through these maps, designers can visualize how knowledge spaces interact across cyber, physical, and social dimensions, and to anticipate which decision policies are likely to expand or shift over time [31] [33]. In this way, the design stage is where adaptation and evolution are intentionally structured into the architecture, with indicators guiding the coverage and depth of preparation.

Once designers deploy the system, we move our focus from anticipation to governance. We now use the indicators observationally, and through them we obtain feedback on how the system evolves in operation. We measure responsiveness as the reaction time to disturbances. We see stability in whether adjustments persist across cycles. We track retention by recording which adaptations survive across multiple states.

At runtime we must also log decision provenance continuously so that a designer can audit the adaptations. Equally, indicators of legitimacy and trust must be validated in practice, not assumed. For example, ongoing monitoring of stakeholder satisfaction, scenario walkthroughs, and simulation of emergent conditions can help determine whether evolution is proceeding in a way that maintains both resilience and social acceptance [31, 33] [34]. In this sense, the runtime stage is less about design foresight and more about continuous evaluation and course correction.

When designers view adaptation and evolution through indicators, they change how they design evolving cyber-physical-social systems in three ways. In post-disciplinary work, researchers in no single discipline can fully prescribe what counts as a "good" adaptation or a "successful" evolution. The integration of engineering, social sciences, and cognitive sciences requires a shared vocabulary of observables that each domain can recognize and act upon. We use indicators in this role: through them we make change visible across disciplinary boundaries and judge system trajectories collectively.

At the design stage, this means engineers, social scientists, and human factors experts can converge on preparedness metrics that capture both technical resilience and social legitimacy. For example, simulation coverage or diversity of scenario sets can be evaluated not only in terms of system performance but also in terms of how well they anticipate stakeholder perspectives. Indicators thus become boundary objects that enable different disciplines to contribute to the same design conversation [29, 31].

During deployment we use the indicators for governance. By monitoring responsiveness, stability, retention, and legitimacy at runtime, practitioners from diverse backgrounds — operators, policymakers, and community stakeholders — share a common basis for deciding whether they find the adaptation acceptable and the path of evolution sustainable. This moves validation away from narrow technical verification and toward a broader, transdisciplinary notion of assurance that integrates efficiency, resilience, trust, and human acceptance [27-29]. Finally, indicators shift the focus of post-disciplinary design from producing fixed artifacts to cultivating evolving systems. By treating adaptation and evolution as processes to be continuously measured and guided, design becomes reflexive: open to revision, accountable to stakeholders, and capable of incorporating new knowledge spaces as they emerge. This creates the conditions for design practice that is both rigorous and humane, where technical excellence and social legitimacy are not competing priorities but mutually reinforcing goals.

2.1.6 Micro, Meso, and Macro Scales of Adaptation

To explain the relationship between adaptation and evolution, we categorize adaptations into three nested time and impact scales:

- Micro-adaptation involves rapid, localized changes such as parameter tuning, rule updates, or behavioral shifts in system agents. These adaptations typically occur over a range of milliseconds to hours and are aimed at resolving short-term disturbances or optimizing local performance. Examples include modifying sensor sampling rates based on signal, noise or rerouting communication to avoid temporary failures. Micro-adaptations are reactive and often invisible from a system-level perspective.
- Meso-adaptation refers to intermediate-scale reorganizations such as reassigning system roles, reconfiguring subsystems, or adjusting workflows. These changes occur over days to weeks and involve semi-structured modifications that may span multiple subsystems. Through meso-adaptation we often adjust to persistent shifts such as sustained changes in usage patterns or resource availability. While still reversible, these adaptations require coordination and reallocation of resources.
- Macro-evolution is used to describe deep, long-term transformation in the system's architecture, goals, or identity. Occurring over months to years, macro-evolution reflects a redefinition of the system itself. For example, a system originally designed for emergency

response might evolve into a community service platform due to stakeholder feedback, external partnerships, or new regulatory demands. Macro-evolution is often the result of sustained micro and meso adaptations that are retained and institutionalized within the system.

2.1.7 The Amplifying Role of Social Space

In E-CPSS, the social space is not just as an information source but as a dynamic evaluator of adaptations:

- Stakeholders keep the adaptations they trust and abandon those they do not. When a change matches what users expect, they retain it. When it makes them distrustful or dissatisfied, they may reverse it even when it performs best against the technical measures designers set.
- Stakeholders who grant or withhold legitimacy accept or reject structural reconfigurations. For example, decentralizing decision-making might improve resilience; however, if stakeholders perceive it as a loss of transparency or accountability, they may resist it.
- Through their collective goals, stakeholders shape the trajectory of adaptation and steer the system toward outcomes they endorse. Systems evolve not only through internal logic but also through external social pressures and values. For instance, shifting from energy efficiency to equity-focused energy distribution is an evolution driven by changing social priorities.

2.1.8 Method of the Study

Through this position chapter, we provide a structured perspective on how E-CPSS should be conceptualized and designed from a decision-based standpoint. The primary aim is not to conduct a systematic literature review in the traditional sense, but to construct a coherent conceptual foundation through targeted scoping of the literature, appropriate to support and justify the theoretical stance adopted in this work.

Purpose and Approach

To formulate a clear and actionable position, we undertook a scoping review of academic literature across the domains of cyber-physical systems, complex systems, and socio-technical system design. The objective of this review was to:

- Survey the current landscape of system-level change in CPSS,

- Identify recurring principles and terminology related to adaptation and evolution, and
- Put forth conceptual and methodological gaps that inform our stance on the design of evolving systems.

We use this scoping review as conceptual grounding for the position we articulate in the latter sections of the chapter, including potential frameworks for E-CPSS modeling and our proposed cognitive model.

Core Question Under Consideration

We pose the following central question: *How do Cyber-Physical-Social Systems (CPSS) change over time?*

Through this question, we guide the inquiry into mechanisms of change, both short-term and long-term, within CPSS. It shaped both the scope of the review and the selection of conceptual building blocks that underpin our position.

Search Strategy

Our search spanned academic databases, including IEEE Xplore, ScienceDirect, SpringerLink, and Google Scholar. The following combinations of search terms were used:

- Keywords associated with system change: adaptable, self-adaptable, flexible, evolving, self-organizing, robust, resilient, context-aware, interoperable, decentralized, emergence, scalable (including root variations)
- Combined with system types: Cyber-Physical Systems (CPS), Cyber-Physical-Social Systems (CPSS), and Social Systems.

We also performed citation analysis of two seminal works in the CPSS space:

- Abera and co-authors. (2020): “A New Paradigm and Meta-Model for Cyber-Physical-Social Systems” [1]
- Wang (2010): “The Emergence of Intelligent Enterprises: From CPS to CPSS”[11]

These works are intellectual anchors for distinguishing CPSS from earlier or more narrowly defined system categories.

Selection Criteria and Screening

To keep the review focused and relevant to the position we are building, the following criteria were applied:

Inclusion Criteria:

- Works published after the year 2000
- Studies that explored evolving, adaptive, or resilient characteristics in CPSS
- Papers that contributed to understanding decision-making, system structure, or human-in-the-loop dynamics in complex systems

Exclusion Criteria:

- Papers including CPS without a social component were used for background but excluded from the primary synthesis
- Studies focused exclusively on social change without integration into CPSS frameworks were removed
- Pre-2000 publications were only included if they remained influential.

Review of Outcome and Role in the Chapter

From over 120 initially identified sources, 50 papers were retained after applying the above criteria. These informed:

- The articulation of functional, behavioral, and structural requirements for E-CPSS
- The development of our micro-meso-macro adaptation model
- The formulation of the cognitive map that illustrates decision-making and evolution in E-CPSS
- The identification of conceptual gaps where our position offers value, especially around the role of social cognition and multi-scale adaptation

Through this process, we ensure that the position advanced in this chapter is both grounded and original, addressing a specific void in how system evolution is conceptualized in design literature. Through this review, we justify and clarify our position on how designers should

engineer E-CPSS with adaptive and evolutionary capabilities across various time scales and social contexts.

2.1.9 Towards an Implication for Design

To design E-CPSS that are not only adaptive but evolution-ready, the following design strategies are essential:

- Track and model adaptations across time scales

  Researchers should examine bias detection and monitoring, adaptive mitigation strategies, and continuous feedback as early design considerations in the design of an E-CPSS.

- Integrate stakeholder feedback loops into adaptation logic

  Researchers should engage social agents as real-time evaluators, who offer explicit or implicit signals of whether an adaptation maintains functional adequacy as well as trustworthiness and acceptability. Researchers should log this feedback as labeled events and analyze it alongside disparity metrics to guide mitigation or retraining when concerns arise.

- Recognize thresholds beyond which adaptation becomes evolution

  When researchers observe that changes in behavior and structure persist across iterations and begin to reshape a system's identity or purpose, researchers should recognize a state transition, that is, system evolution. Researchers should use the Requirements List in Section 2.3.3 to frame decision variables, constraints, and stakeholder commitments that guide adaptation within each state. Researchers should use the Cognitive Map in Section 2.4 to trace information flow and decision points across the cyber, physical, and social spaces, and to plan how adaptations accumulate into evolution. Researchers can use these two instruments to support decision based design that structures adaptation pathways and maintains technical fidelity, social legitimacy, and cognitive alignment across states.

We apply this three-tiered adaptation lens when critiquing prior work. We propose the joint use of a Requirements List and a Cognitive Map to structure adaptation pathways and to stage indicators by state. The authors claim that this pairing enables traceable legitimacy checks in parallel with technical checks and extends current guidance on CPSS design.

In the remainder of this chapter, we conduct a systematized literature study, focusing on the design aspects of an E-CPSS. This involves various scholarly works concerning the evolution of CPSS,

systems design, modeling paradigms, system characteristics, the Function, Behavior, and Structure (FBS) construct, and decision-support mechanisms. Through this scoping examination of literature, the authors synthesize insights into the dynamic nature of the E-CPSS, elucidating the intricate interplay between system components, evolving functionalities, various modeling approaches, and decision-making mechanisms through a cognitive map. Key characteristics of evolving CPSS are presented, and decision-support frameworks essential for addressing dynamic changes within these complex systems are delineated. In this position chapter, the authors have leveraged the existing knowledge of the literature to present further avenues for exploration and put forth critical research findings. We apply this lens in the literature review (Section 2.2) and the requirements synthesis (Section 2.3).

2.1.10 Verification and Validation

Having distinguished adaptation from evolution and introduced indicators that govern both design-time preparation and runtime governance, we now specify a verification and validation approach tailored to E-CPSS. Because objectives, stakeholder legitimacy, and error regimes can change across states, verification and validation must proceed iteratively: designers log decision provenance, re-verify at each state transition, and rely on stakeholder inquiry questions when fixed requirements no longer apply. We complement stage-bounded quantitative checks with social-model assessments and external accreditation to maintain acceptability as the system evolves.

Bias-aware design for an E-CPSS can draw upon three linked practices. At design-time, designers may pair sampling plans with stakeholder stratification, audit label definitions with domain experts, and stress-test data pipelines with counterfactual profiles before deployment. During run-time, designers may monitor disparity metrics across salient groups and log human overrides for periodic adjudication. When pre-agreed thresholds are exceeded, designers may adapt the system through reweighting or retraining, and convene stakeholder review when high-impact decisions display recurring disparities.

As with all models acting as stand-ins for reality, it is appropriate to consider methods for verification and validation. While all models and systems should strive for overarching verification and validation standards using the standard lexicon, E-CPSSs present unique challenges. General strategies remain the same, for example, the assessment of design considerations is validated, and

a reduction in error is sought during iterative verification throughout the process. However, both evolutionary models and social systems have unique considerations. Thankfully, the interactions of these unique considerations are synergistic.

To assess verification and validation in an evolving system, researchers adapt methods to the current system state [35]. Because an E-CPSS changes over time, researchers reassess validation iteratively to judge whether stakeholders' goals and inquiry questions remain satisfied within each state [35], [36]. Researchers verify in stages and they recognize that error measures that hold within one state may not transfer to the next [35]. When emergent properties appear, researchers evaluate the tolerance for shifts in error and they decide whether those shifts merit system modification [36]. Researchers also apply statistical tools, such as cross correlation analysis and signal resolution, to separate signal from noise within a state [37]. With these practices, researchers close a simple control loop. During design, researchers declare thresholds and inquiry questions. During operation, researchers compare current indicators with declared thresholds and then decide whether to remain in the current state, to adapt within the state, or to initiate a state transition

With indicators in place as explained in Section 2.1.5, researchers can stage evaluation across design time and run time, drawing on simulation, agent based modeling, and stakeholder scenario testing [38]. For example, researchers can use a simple agent based market with two stakeholder groups to test whether a planned policy update reduces disparity without reducing stability in the next state. During design, researchers frame inquiry questions with stakeholders, define state specific metrics, and use agent based testbeds to explore counterfactual scenarios [38]. These preparations support sensitivity sweeps, scenario stressors, and bias audits before deployment [38]. During run time, researchers re verify at state transitions, monitor group wise disparity continuously, and adjudicate human overrides [38]. After major transitions, external reviewers may assess whether the system continues to meet standards of legitimacy and acceptability [39].

Because these activities concern social behavior, researchers recognize the limits of empirical testing for social models [37]. Researchers reduce the risk of subjectivity by pairing verification and validation with external criteria and by reporting proximal metrics transparently [37], [38]. In the next paragraph, researchers describe how outside accreditation complements verification and validation [39].

To assess the verification and validation of an evolving system, the methods must also be adapted to the system state. Since an E-CPSS is perpetually changing, validation must be iteratively reassessed to ensure requirements are met [35]. This is somewhat complicated by the nature of an evolving system; system requirements may not persist throughout evolution. Focusing on goals and inquiry questions instead of requirements addresses this problem [36], administered as timely stakeholder check-ins. Similarly, verification should proceed in stages. While tried-and-true methods like RMSE analysis are valid, they may only be valid within a particular stage. As emergent properties arise, the error may undergo a radical shift to the new paradigm. The role of the designer should be to assess the tolerance for such error shifts and to determine if such uncertainty merits system modification. This may lead to a greater reliance on cross-correlation analysis, signal resolution, and similar statistical tools that can differentiate signals from noise.

Social models are notoriously resistant to empirical testing [37]. Designers must be aware that verification and validation for E-CPSSs are susceptible to a loss of objectivity without external criteria. Instead, designers must address proximal metrics of verification and validation, such as checking internal system logic and verifying data without external standards. Assessment of social systems relies heavily on stakeholder input through techniques such as subject matter expert interviews, stakeholder surveys, and credibility assessments, as well as tools familiar to agent-based model practitioners, including invalidation exercises, compositional consistency checks, and process validation [38]. Researchers should name resource and governance constraints that shape design choices. Examples include computational budget versus monitoring coverage, operator labor versus frequency of stakeholder debriefs, and privacy risk versus data richness for disparity analysis. Researchers should record chosen tradeoffs in the design dossier and should revisit those tradeoffs at each state transition. Accreditation, in addition to verification and validation, can use multiple outsider perspectives to determine if the model is "consistent, stable, and repeatable" by ensuring consensus of reviewers [39].

In practice, verification and validation of an E-CPSS may draw upon a minimal set of complementary measures. Technical fidelity can be judged through prediction error and stability indices applied to the cyber-physical functions within each state, using methods familiar from systems engineering [37]. Social legitimacy can be assessed through disparity analyses, evidence of stakeholder acceptance, and opportunities for redress, supported by stakeholder surveys and

credibility assessments [38]. Cognitive alignment can be examined through the adequacy of explanations as experienced by stakeholders, with adequacy tested in debrief interviews and scenario evaluations. These measures can be staged by system state, with tolerances reassessed as transitions occur to maintain validity across time.

Despite efforts to address social model uncertainties, significant gaps remain in social model assessment. While future research may create or enhance existing tools, the ultimate question of assessing a model of a social system should be, 'Does it serve?' This is synergistic with evolutionary systems. The best practice for a challenging system is for the designer to continually reassess the system. Conveniently, an evolving system must be continually reassessed, as verification and validation from a previous system state may no longer be applicable. Much like the evolving system may be separated into conceptual holistic states in Figure 2.2, verification and validation assessment should accompany each state. A designer should assign trust scores to verification and validation from previous states as a priori state history, thereby informing the assessment of the next state.

We organize the chapter as follows. Our objective in Section 2.1 is to provide the readers with the foundational knowledge required to understand the necessity of CPSS and the hierarchical evolution of CPS to CPSS. Furthermore, in Section 2.1, we also provide a brief overview of the evolution of systems, which further delves into the fundamental concept of E-CPSS. As mentioned previously, we explicitly outline our objectives throughout this chapter. In Section 2.2, we provide a detailed description of social space and address its significance to E-CPSS. We also address its utility for decision support in complex systems and the evolution of social entities. In Section 2.3, we discuss the state of the art of E-CPSSs and the most significant research questions discussed in the literature as we define E-CPSS. In Section 2.4, the characteristics of E-CPSS are shown and further classified into a detailed requirements list based on Function, Behavior, and Structure. In Section 2.4, we discuss how designers use the cognitive map during E-CPSS decision making, and we examine the frameworks, models, and methods that researchers report in the literature. Then, we present a cognitive map of an E-CPSS based on the knowledge of the state of the art, the characteristics, and the requirement list established in Section 2.4. In Section 2.5, key findings are presented, along with key avenues for research that address gaps in the context of fundamental knowledge required, the E-CPSS paradigm, and Design Engineering. Then, in Section 2.6, closing

remarks are presented, along with key takeaways for the paradigm of decision-based design support for E-CPSS. In the Appendix, we provide a brief description of some of the methods which have been used to model complex systems.

An evaluation process for an E-CPSS can be staged across design-time and run-time activities, drawing on simulation, agent-based modeling, and stakeholder scenario testing. During design, researchers may frame inquiry questions with stakeholders, define state-specific metrics, and use agent-based testbeds to explore counterfactual scenarios. These preparations allow sensitivity sweeps, scenario stressors, and bias audits to be conducted before deployment. During run-time, evaluation may include re-verification at state transitions, continuous monitoring of group-wise disparity, and adjudication of human overrides. After major transitions, external reviewers may be invited to assess whether the system continues to meet standards of legitimacy and acceptability. Together, these activities outline a pathway for conceptual evaluation without implying a single prescriptive procedure.

2.2 Design and Decision Making In The E-CPSS

Previous authors have considered the design and development of systems over time. For example, design for the circular economy (including design for the life cycle), sample publications include [32] [33], and the development of product platforms/families, sample publications include [40] [41] [42]. These efforts have generally been focused on waste management, cost savings or efficiency. Here, we propose a much more active role for understanding the social space and cognition in the system being designed.

2.2.1 Importance of the Social Space

Researchers view the CPSS paradigm as an important perspective on the social space, which is relatively less explored than the cyber and physical spaces. This includes defining the social entity, addressing the role and impact of a social entity on the system, and advancing the computational capabilities to address the social entities. There are systems such as the Internet of People (IoP), Human-in-the-Loop (HiTLCPS), SIoT (Social Internet of Things), etc. [43, 44] to incorporate the role of a social entity. However, the role of humans in social entities requires clarification. HiTLCPS [45, 46] is a promising framework for integrating humans into the decision-making

process. However, there are problems in identifying the role of humans. Factors such as human cognition, intentions, emotions, and behavioral predictability are loosely defined in a HiTLCPSS framework. Thus, the human reliability assessment (HRA), which is defined by Jirgl and co-authors [38] as, "the science disciplines dealing with the human factor assessment, especially in the view from its reliability." This entails examining human performance in diverse operational scenarios, modeling human behavior, and evaluating its reliability and other attributes.

Although many methods for system design have been proposed, we believe that the Function Behavior Structure approach is ideally suited to help a designer formulate the design of a complex system, such as an evolving CPSS. Problem formulation is an essential part of a successful design, and many of the current design methods provide little help in this area; even the system engineering vee's task clarification stage starts with a well-defined set of desired requirements for a system, which does not evolve and is rather vague when confronted with problems of system evolution. Typically, E-CPSS designers are human, as humans usually specify the initial function during E-CPSS operation. However, decision makers can be either external or internal to the system, and input from social entities can be considered along with input from cyber and physical entities.

A critical challenge in considering inputs from social entities as drivers of system evolution is distinguishing between HiTL interactions and biases resulting from human inputs. The integration of HiTL is a foundational part of system architecture, enabling systems to incorporate preferences and contextual awareness in decision-making. However, direct human input, as one of the social entities, can introduce biases during system design, which can further propagate, resulting in the ignoring of minority viewpoints and giving preference to a specific set of decision-making parameters, based on the weights and biases used during system design (or training data). This, as a challenge, has been discussed as future work in Section 2.6.

There are two major concerns associated with the design of E-CPSSs: i) problem formulation, that is, identifying what is to be done, and ii) the representation of issues related to socialization and cognition. Problem formulation is a matter of uncertainty because it is unclear how the system will achieve its function in changing exogenous and endogenous environments. Understanding the definition of an E-CPSS is essential for designing the systems. To proceed with designing an E-CPSS, either a designer may bound all possible future states, or specify rules for system behavior to meet various eventualities, and further rules may be added. Typical advanced design procedures combine both approaches. For example, consider a cell phone with upgradable software; its

behavior will change with the upgrades. However if a camera is not available in the original camera structure, the cell phone's functionality is bounded by its structure.

2.2.2 Social Space Literature Highlights

The second concern for designing E-CPSS is the representation of social issues, such as trust, coordination, and cooperation, which are integral to socialization and cognition. However, the use of these factors in system modeling is relatively unexplored, despite the development of various models for their assessment. Trust, cooperation, and coordination are often viewed as qualitative constructs; various methods have been proposed for quantifying these constructs in specific situations. Some of these methods are given here, but it is apparent that factors such as these should be modelled in the context of each specific E-CPSS under consideration. For instance, a framework for modeling network trust among supply chain executives using behavioral data has been presented, enabling the quantifiable integration of trust dynamics into decision systems [47]. Similarly, some researchers utilized experimental economics to quantify trust and social distance, illustrating how trust-based interactions can be mathematically modeled based on context and social proximity [48]. Gharib and co-authors [49] discuss the significance of trust through an example of ambient assisted living. The need for trust arises when human and social agents depend on one another, as risks are entailed in dependencies between different humans (agents) [50]. Efforts at quantifying cooperation within organizations include the work of Huybrects and co-authors [51]. They argue that cooperation has four components (valorization, validity, reactivity and retroaction). Di Stefano and coauthors seek to understand the dynamics of cooperation in terms of multiplicity and homophily, that is, the size of an individual's network and their tendency to associate with those who are similar. Nunez and co-authors relate coordination to task uncertainty and interdependence. [52] Marcus and Le propose assessments of cooperation based on individualism and collectivism [53]. When using social constructs such as these to analyze E-CPSS, it is important to choose constructs that are appropriate to specific contexts.

Apart from the preceding discussion on incorporating human cognition and behavioral aspects, the social space also consists of sociocultural elements and the social environment. In the field of social science, Turner [54] defines a social interaction as a "situation where the behaviors of one actor are consciously reorganized by and influence the behaviors of another actor, and vice versa, which can be extended to the process whereby there is a mutual influence between behaviors of multiple

individuals." According to the social action theory proposed by Weber [55] social interaction involves accommodating the actions of others, which, in essence, requires a compassionate comprehension of varied interests, perspectives, opinions, and behaviors. Social interaction is fundamental to the significance of social space. Most social problems like environmental justice, equitable distribution of resources, and healthcare involve humans as the core of social space and the external social environment. For problems like these, it is critical to understand and interpret the behavioral and cognitive aspects of various stakeholders' preferences, opinions, interests, and beliefs, and incorporate these elements in the system's design. When designing an E-CPSS we recommend starting with a top-down approach. Using the Function Behavior Structure approach, a designer can pre-define the desired functionality for all possible states that the system will encounter during its life span and then design a system that can adjust to those circumstances. This is used to determine the desired potential overall lifetime functionality of the E-CPSS. Once the design problem is defined, it becomes a matter of design under uncertainty, and methods such as robust design have been developed to deal with design uncertainty. This approach, when used alone, tends to increase system size, as the entire structure to support the required function must be incorporated into the system on the first day. Another approach to top-down design is to develop a set of rules for system behavior in the context of the desired function. Designers can update these rules as conditions change, and when they also design the system modularly they extend its lifespan at a lower initial investment.

Once the desired function has been determined, a bottom-up design approach is required to develop the structure(s) and behavior(s) to accomplish this function. For many years, we have advocated a design approach called decision-based design, and have recently extended it to the Design Evolution Cycle. [56] [57]. We believe that engineering design is accomplished with a series of decisions, with a successive refinement of the system being designed [58] [59] [60] [61] [62]. This work is based on the ALP algorithm and the compromise Decision Support Problem [63]. This approach has been used successfully for many different kinds of physical and cyber-physical systems, some of them are: [64] [65] [66] [67] [68] [69].

Whether design decisions or operational decisions, information is aggregated into models, which are then exercised to provide information for decision-making. Again, this decision-making may be handled either by humans or the CPSS itself. Designers who apply standard optimization

methods to social problems often fail, because the information available to them conflicts [70]. They face further difficulty when many stakeholders take part. It is critical to consider the interests of as many stakeholders as possible to arrive at satisfactory rather than optimal design options. The inclusion of social entities enables decision-makers to interpret and analyze various scenarios for social problems, thereby accommodating the preferences of the maximum number of actors involved. It also enables us to consider diverse preferences and explore a broad range of feasible solutions for any given problem. Furthermore, including social entities enables real-time feedback through interaction between the cyber-physical and social spaces, making the changing conditions, sharing of information, and decision-making support more dynamic and responsive, thereby aiding in addressing evolving social entities.

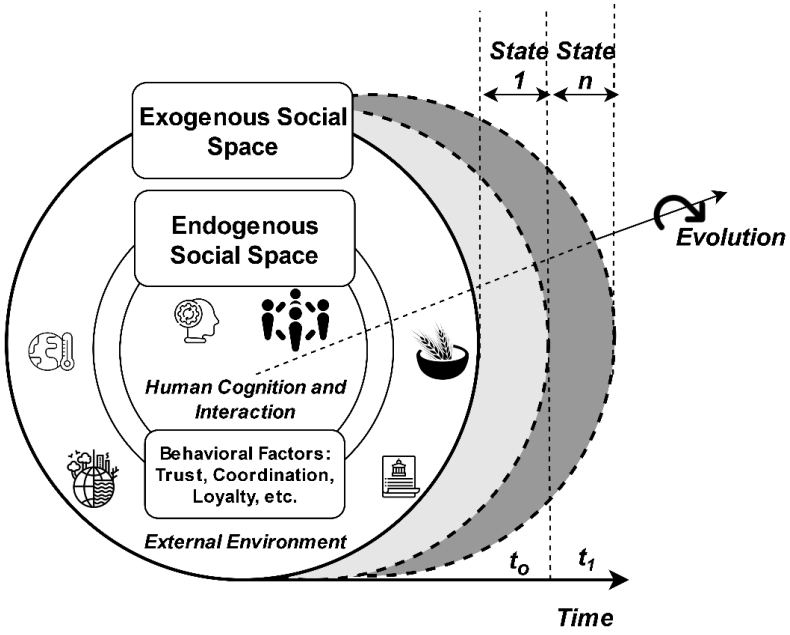


Figure 2.2: Holistic Social Space of CPSS

Including social entities that account for human behavioral and cognitive factors, along with Human Reliability Assessment, enhances the ability of designers to make realistic decisions by incorporating interpretive analysis, thereby enabling more informed decisions in CPSS. For example, understanding the roles and objectives, as well as the preferences, of social entities is critical in designing policies. Further, aiding the roles and objectives, as well as the preferences, of social entities is critical in designing policies, which in turn support decision-making. The term

"social entity" is used with multiple interpretations and encompasses a range of meanings in today's academic and research literature. A few examples of social entities (or related considerations) within current publications on CPSS are presented in Table 2.1, where the social entity examples from various papers are reviewed, along with their corresponding paper numbers from the list of references.

While the preceding discussion is focused on the social space from the perspective of human cognition and behavioral factors, we now provide some insights into an integral part of social space, under consideration, namely, the environment. When considering social space as a combination of both endogenous and exogenous components, the cognitive, behavioral, and interactive dynamics among system stakeholders can be examined as endogenous components. In contrast, the social environment of the CPSS under consideration can be viewed as an exogenous component consisting of broader societal forces, cultural norms, and environmental changes external to the immediate system, as shown in Figure 2.2. The external environment that evolves over time is critical for robust design options in CPSS, particularly when the system transitions from State 1 to State n. These considerations empower designers to assess the impacts of social entities on the system by incorporating both exogenous and endogenous social spaces. We recognize that external, continually transforming conditions shape the dynamics of CPSS. Understanding and adapting to these external changes is critical in exploring robust design strategies that can effectively address the multifaceted challenges encountered within the CPSS. As the external environment shifts and adapts, it introduces opportunities and threats to the stability and functionality of CPSS. Strategically navigating these external forces becomes a paramount consideration for social space. In this context, socialization plays an important role in shaping how human agents interact with one another and with the system. Within the endogenous social space, socialization manifests through patterns of stakeholder interaction and the development of social behavioral constructs like trust, loyalty, etc. In parallel, researchers observe that external conditions shape socialization processes, which in turn influence how stakeholders adjust their behaviors and priorities . As shown in Figure 2.2, changes in environmental factors also lead to changes in socialization processes, which redefine stakeholder roles, preferences, and decision-making logic over time.

Table 2.1: Examples of Social Entities

Social Entity	Paper
Workers and technicians and their interaction with one another and the CPSS object that is, a cobot	[1]
Interaction with inhabitants and other CPSS objects	[11]
Users and policymakers - plant in charge, consumers	[71]
Humans and their associated interaction with cobots. The features of personalization and socialization are imbibed in smart devices.	[72]
Human beings are sensors and users of technology. Social media as an information source on traffic routes	[4]
Humans interacting and using the social network	[23]
Self-organizing capability through obstacle detection and collision prevention	[8]
Self-organizing capability according to the environment.	[73]
Situation: humans as data sources	[74]
Customer and Community Engagement	[75]
Social behavior and social organizations	[76]

Passengers, drivers, government	[77]
Learning and training of workers, Change in roles and objectives of workers due to introduction of CPS.	[78]
Nurses, patients, doctors	[79]

2.2.3 Evolution in Social Space

A social entity is dynamic, meaning it changes over time. Human cognitive and behavioral aspects constantly evolve in response to the requirements, preferences, and changing goals influenced by the changing circumstances in the external social environment. Changes in the social sphere necessitate traditional cyber-physical Systems (CPSs) and push them toward becoming CPSSs. At the same time, the demand for increasingly complex functionality pushes systems to evolve. The importance of considering evolution is to guide how systems should adjust in response to changes in external stimuli, which is fundamental for designing E-CPSS. For example, in Figure 2.3, we present the effect of dynamic social environments, both endogenous and exogenous, on decision strategies as shown by network structures. As the problems change through different states, the network structure evolves, inherently making the decision strategies differ concerning time states. As the problem progresses across various states, designers must recognize distinct decision strategies rooted in the diverse network structure corresponding to each facet of the problem, as illustrated in Figure 2.3. For example, Problem 1 will have a different set of network structures that includes specifics regarding the interaction of stakeholders and the social environment. This leads to the design of a system anchored in Decision Strategy 1. However, at a different stage of time, as shown in Figure 2.3, Problem 1, which evolves into Problem 2, has a different decision strategy owing to a different network structure. Thus, the design of E-CPSS caters to the requirement of being self-adaptive by considering different aspects of social spaces, as shown in Figure 2.2. Thus, the evolution of CPSS in general and the social entity, in particular, is critical for the decision support of the systems and provides predictive abilities to system design by leveraging the ability to make real-time decisions for complex systems. These evolving decision strategies

reflect adaptation through dynamic interaction rules, akin to evolutionary computation in complex adaptive systems [47].

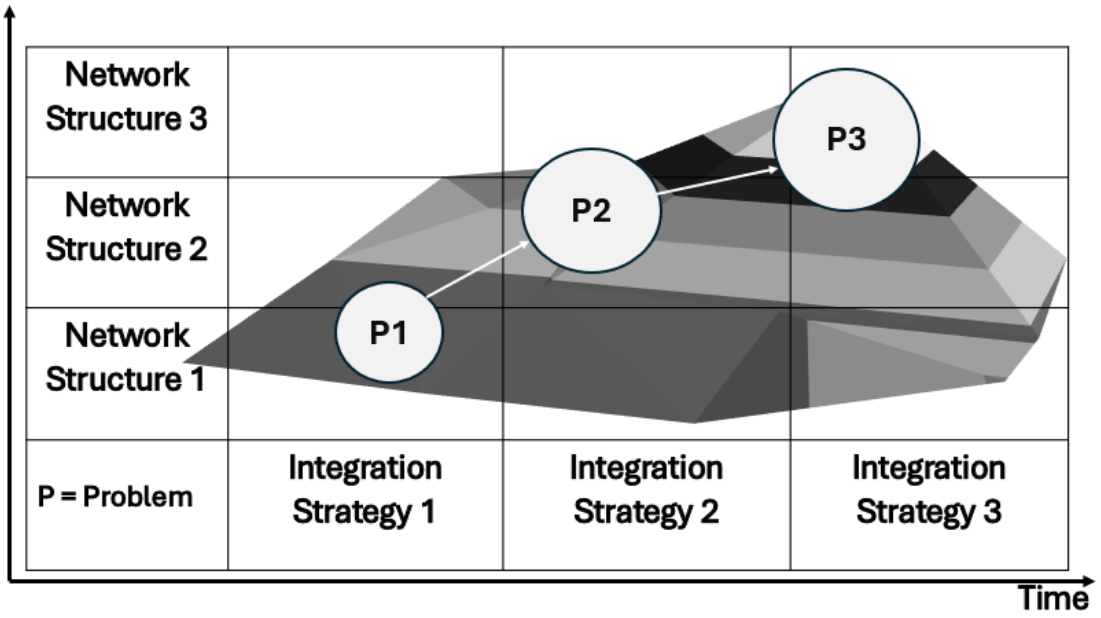


Figure 2.3: Evolving Problem Structures and Decision Strategies

2.3 EVOLVING Cyber-Physical-Social System

2.3.1 Summary of State of the Art and the Foundational Insights for an E-CPSS

We do not focus on experimental studies; instead, we offer readers a few examples of practical CPS implementation. In Table 2.2, we summarize our key findings and supporting references based on the systemized process of studying the literature to provide foundational insights into an E-CPSS by discussing several factors and requirements that can help designers design such systems. By furnishing a literature summary, we have also identified research gaps in the state of the art discussed in this section of the chapter. We classify our findings into four categories: evolving characteristics, modeling and frameworks, causes of evolution, and approaches and methods. We use the evolving characteristics that are identified and summarized in this chapter, which are self-adaptability, self-organizability, resilience, robustness, flexibility, scalability,

decentralized decision-making, and emergent behavior. Potential models and frameworks are identified and categorized in Table 2.2 to model such characteristics while designing an E-CPSS. These models and frameworks are mainly ACP (Artificial societies, Computational experiments, and Parallel execution), Parallel World, Knowledge Automation, Force Field Control, and Mape-K, and some models that do not fall in the above category, are explicitly mentioned, such as GAN (Genetic Adversarial Networks), ADACOR (ADAptive holonic COntrol aRchitecture) and MBSE.

Similarly, approaches and methods discussed in the state of the art in the context of the evolving nature of an E- CPSS are listed and categorized into artificial societies, parallel execution, human dynamics, scenario-based, and social computing. Finally, we name the stimuli that lead to evolution and for which designers must design: changing stakeholder needs, feedback, uncertainty, risks and disruptions, and unwanted emergent behavior. Building on Section 2.2, we emphasize the social entities and how designers change a system as conditions around it change. Drawing from Table 2.2 and the literature study, we present the research gaps in the state of the art.

From Table 2.2 we infer that researchers address self-adaptability, interoperability, emergence, context awareness, robustness, resilience, and decentralized decision-making most often, and that they leave self-organizability, scalability, and flexibility comparatively unexplored. Researchers have not yet developed a uniform framework through which designers model and design for an E-CPSS and its evolving characteristics. We therefore state our first research gap:

RG1. There are no frameworks or models to support the design for evolving characteristics and all the stakeholders' needs in an E-CPSS.

Cognition is an important capability of systems undergoing evolution to realize the system's current and evolving state. From Table 2.2, the discussion of cognitive capabilities in the system is underexplored in the state of the art, which leads to our second research gap that is:

RG2. Lack of fundamental knowledge to realize collective advanced cognitive capabilities in a system causing evolution.

Changing stakeholder needs, feedback, and interactions from the social entities are inferred to be the most common factors responsible for causing evolving characteristics in the system, which leaves the role of other factors in bringing about evolution underexplored. Additionally, the discussion of human dynamics and its role in bringing about evolution needs to be revised in the state of the art, which should be leveraged to explain the dynamic behavior in the system caused

by social entities. Additionally, a thorough discussion on different social integration strategies must be included, leading to our third and fourth research gaps as follows:

RG3. Role of social space and human dynamics in bringing about evolving characteristics in the system and the effect evolution has on social entities.

RG4. Finding the different social integration strategies to design for social dynamics in an E-CPSS.

In an E-CPSS, the cyber, physical, and social entities operate in synchronization to adapt to the external environment collectively. As a result of this adaptation, emergent behavior is observed that can act as uncertainties and risks for the system. It can be seen from Table 2.2 that the discussion in collective adaptation and the corresponding uncertainties arising in the system is untapped. It can be seen from the column "Causes of Evolution" in Table 2.2 with data corresponding to the various research papers in the column "Uncertainty" and "Unwanted Emergent Needs" that the research in this area is unexplored which leads to our fifth research gap that is:

RG5. Dealing with uncertainties, disruptions, risks, and emergent misbehavior due to collective adaptation and evolution.

Through the research gaps identified, we highlight areas for enhancement in this field of fundamental research gaps. These research gaps are summarized based on the insights obtained from Table 2.2, representing the sufficiency of various attributes in the state of the art. We use these findings to address research gaps (RG1 - RG5) and provide depth to the current study on an E-CPSS. Drawing insights from the information in Table 2.2, we now provide definitions for CPSS and E-CPSS in Section 2.3.2. In these definitions we take account of the aspects we draw from those papers, and we treat social entities and evolution as design dimensions. Additionally, we leverage the insights gathered from Table 2.2 to devise a cognitive map for the E- CPSS in Section 2.4 to explain its evolutionary mechanism when subjected to an uncertain environment and its evolving attributes.

Table 2.2: State of the Art and Foundational Insights on E-CPSS

* S- Social Entity; E - Evolution

	EVOLVING CHARACTERISTICS											MODELING AND FRAMEWORK					CAUSES OF EVOLUTION					APPROACH/ METHODS					S *	E *
PAPER	Self-Adaptability	Self-Organizability	Robustness	Resilience	Emergence	Scalability	Flexibility	Interoperability	Decentralized Decision	Modularity	Context Awareness	Parallel World	ACP	Knowledge Automation	Force Field Control	MAPE-K	Changing Stakeholder	Feedback/Closed Loop	Uncertainty	Risks/Disruptions	Unwanted Emergent	Artificial Societies	Parallel Execution	Human Dynamics	Scenario Based	Social Computing		
[1]	*		*	*	*	*		*			*	*					*	*	*			*	*		*	*	Y	N
[11]	*	*	*		*		*	*			*																Y	N
[8]	*		*	*	*	*		*							*			*			*						Y	N
[19]	*		*		*																						Y	Y

[74]	[81]	[78]	[73]	[4]	[4]	[80]	[76]
*				*	*		*
*			*				*
*			*	*	*	*	
*				*		*	*
*	*	*		*	*	*	*
	*			*		*	*
*				*			*
*	*	*	*	*	*	*	*
	*						
					*		*
*	*	*	*	*	*		*
				*	*		*
					*		*
*	*				*		
		*		*		*	*
*	*	*		*		*	*
						*	
							*
							*
					*		*
					*	*	*
*				*	*	*	*
*	*	*	*	*	*	*	*
				*	*		*
Y	Y	Y	Y	Y		Y	Y
N	Y	Y	Y	Y		N	N

[82]	*				*			*			*			*													Y	Y
[83]	*	*	*		*			*			*			*			*		*					*			Y	N
[84]	*	*	*		*						*																Y	N
[21]	*	*			*			*			*	ADACOR													*		N	Y
[43]	*		*	*	*			*	*	*	*	*					*	*			*		*	*	*	*	Y	N
[85]	*	*			*		*	*		*	*						*	*				*		*	*	*	Y	Y
[86]		*			*		*	*				ADACOR					*	*		*							Y	Y
[87]	*		*		*			*			*	Azure Cloud Framework															N	Y

[2]	*											MBSE					*			*				*			Y	Y
[88]	*			*													*		*	*				*			Y	Y
[89]	*	*	*		*			*	*		*	FBS Design Model					*	*									N	Y
[75]	*	*			*			*	*		*																N	Y
[90]	*	*			*		*	*	*	*	*	CPHS					*								*		Y	Y
[7]				*			*				*	STAMP							*	*				*			Y	Y
[91]						*				*		FBS Design Model															N	N
[92]								*			*	Information Modeling													*		N	N

[93]							*	*			*	Stepping Structure IM					*								*	*		Y	Y
[9]		*			*			*	*		*															*			Y	Y
[94]	*	*			*			*				VAR + GPR					*	*		*					*	*	*	Y	Y
[95]	*				*			*			*						*	*		*					*	*	*	Y	Y
[96]	*		*	*	*			*	*	*	*						*	*										N	Y
[97]		*			*			*	*			HBRGM						*		*	*				*			Y	Y
[98]			*	*	*			*	*		*	Group Value Concept						*	*	*	*				*			Y	Y
[99]				*	*			*	*		*		*				*	*	*	*	*		*		*	*		Y	Y

[100]	*		*	*	*		*	*	*		*	FBS Design Model						*		*								N	Y
[101]	*			*	*		*	*	*		*	GAN					*	*		*				*	*		Y	Y
[102]	*	*	*		*	*	*	*	*		*	Self-Synchronization						*	*	*				*	*		Y	Y
[79]	*				*			*									*											
[49]				*				*	*								*	*		*	*			*	*		Y	Y
[77]	*				*		*	*			*													*	*		Y	Y
[103]	*			*	*		*	*	*		*	FBS Design Model					*	*	*					*	*		Y	Y
[104]	*			*	*			*	*	*	*	Agent Based Models						*							*			

[105]	*		*	*	*			*	*		*						*						*	*	*		Y	Y
[106]	*		*	*	*	*		*	*			Digital Twin						*			*		*				Y	Y
[107]	*	*			*	*	*	*	*		*	Agent Based Model												*	*	*		
[108]	*				*		*	*	*		*	Agent Based Model					*		*	*	*	*		*	*			
[23]	*		*	*	*		*	*	*		*	Collective Adaptation															Y	Y
[112]	*	*		*	*	*					*	Collective Adaptation					*	*	*		*			*	*	*	Y	N
[109]	*	*			*	*	*	*	*																			
[110]	*		*	*				*	*		*					*											N	Y

[71]	*	*	*		*	*	*	*	*		*							*	*		*	*	*		*	*	Y	Y
[72]		*	*	*	*			*	*		*	Social Force Model														*		Y	N
[111]	*	*			*		*	*		*							*	*	*	*								Y

2.3.2 Characteristics and Functional Requirements

When designers include the social space in a CPS, they take on more complexity and uncertainty, and the structure and behavior of the system then change more often. CPSS operates and functions in an open, complex, and dynamic environment wherein uncertainties are a part of the process. The uncertainties introduced by system components and their corresponding environment can disrupt the system's functioning, thereby compromising its performance capabilities. These uncertainties can take the form of changing personal needs, preferences, external intended risks, and unwanted emergent behaviors. When designers integrate the social space, they include social entities and their preferences in the decisions they make, and they then face a wider range of uncertainties. In response, the system must evolve by changing its structure and behavior to adapt to these changes and achieve a steady state. As stakeholders change their needs and preferences, designers adapt the system and produce design solutions that match those needs. An E-CPSS can detect uncertainties and complexities and adapt to an evolving environment by mitigating and managing these uncertainties through changes to the behavior and structure of the CPSS architecture and the associated network, ultimately achieving a steady state. An E-CPSS exhibits various characteristics, such as scalability, flexibility, resilience, and interoperability, which enable it to adapt to these changes. We draw on the literature to identify the characteristics a system needs in order to change with its environment, and we use them to inform the design of an E-CPSS.

As entities that were initially considered to be exterior entities of the system engaging with the CPS, humans are now viewed as a diverse source of information. This has led to the development of the Human in the Loop (HiTL) paradigm. Over the past decade, various researchers have employed different terminologies to describe this integration of the human aspect. For instance, a Cyber-Physical-Human System (CPHS) is defined as “a system of interconnected systems ’talking’ to each other across space and time, and allowing other systems, devices, and data streams to connect and disconnect” [71]. Researchers who work on human-cyber-physical systems treat human elements as separate physical entities, which cyber systems then control and coordinate.

On the other hand, the term Social-Cyber-Physical-Systems (SCPS) used for , “complex socio-technical systems, in which human and technical aspects (CPS) are massively intertwined” [72]. Keeping humans restricted only to a physical entity and considering the social intangibles has led to a separate 'social' space. The term "Social" carries a broader meaning as it reflects a human's emotional, cognitive, and behavioral aspects., Abera and co-authors [1] define CPSS as a "Cyber-

Physical-Social system (CPSS) is a system comprising cyber, physical, and social components, which exists or emerges through the interactions between those components. A CPSS is comprised of at least one physical component responsible for sensing and actuation, one cyber component for computations, and one social component for actuating social functions [1].

Some systems and entities with characteristics similar to those of an E-CPSS, which are undergoing state-of-the-art research, are "Collective Adaptive Systems (CAS)" and "Evolvable Systems." These are systems and entities that show evolutionary behavior over time by adapting to their external environment and accomplishing the objectives for which they are designed [23, 83]. The entities in CAS are synchronized to mitigate and manage any uncertainty arising in the system. As discussed in Section 2.3.2, three conceptualizations (SCPSs, CPSSs, and CPHSs) have surfaced in relation to CPSSs. These conceptualizations differ in that CPSSs can include human stakeholders in the system's operational/servicing loop, while SCPSs aim to use the "system-in-the-loop" (SitL) method by making systems self-adaptive to humans [6]. On the other hand, CPHS involves the smooth integration of human actors into cyber-physical systems, such as autonomous driving or surgical robotics, where human decision-making, cognitive inputs, and interactions are essential to the system's operation. As is common in smart city applications, CPSS broadens this reach by integrating social dynamics and collective behavior into cyber-physical systems, acknowledging the significance of social input, communities, and societal norms in system performance. Meanwhile, SCPS centers on systems where social entities are not just components but actively shape cyber-physical interactions through emergent behavior, network effects, and societal-level coordination, often seen in crowd-sourced traffic platforms or energy cooperatives. While overlapping, these frameworks differ in emphasis, as shown in Figure 2.4: CPHS prioritizes individual human interaction, CPSS emphasizes social structures, and SCPS highlights socially emergent cyber-physical integration.

Humans in the social space of a CPSS act as both “sensors” and “users” of the system . Humans act as "sensors" of the system by providing information consisting of preferences and needs with a certain degree of personalization from every stakeholder. This information, along with sensory cyber-physical information, is further used by the system to understand the context in which the decision must be made to provide design solutions. On the other hand, humans act as "users" of the system as they use the solutions provided to formulate policies and take necessary action. Thus, looking for reliable integration strategies of the social space in a CPSS during the design process

of a CPSS is vital. Concurrent designing and information modeling of a CPSS architecture is of immense importance to the designer, along with social integration strategies to understand the stakeholders' behavior, emotions, and cognition before designing for CPSS. This approach accounts for possible complexities and uncertainties that might arise in the system.

Although various definitions of a CPSS exist, a uniform definition of an E-CPSS still needs to be improved as research into the evolutionary and dynamic behavior of a CPSS is still in its infancy. We define an E-CPSS as

"*An E-CPSS is a system consisting of at least one cyber, physical, and social component which exists or emerges due to the interaction between the tri-space leading to the changes in the behavior and structure over discrete states to achieve a defined function which may or may not change concerning the states in an uncertain and complex environment*."

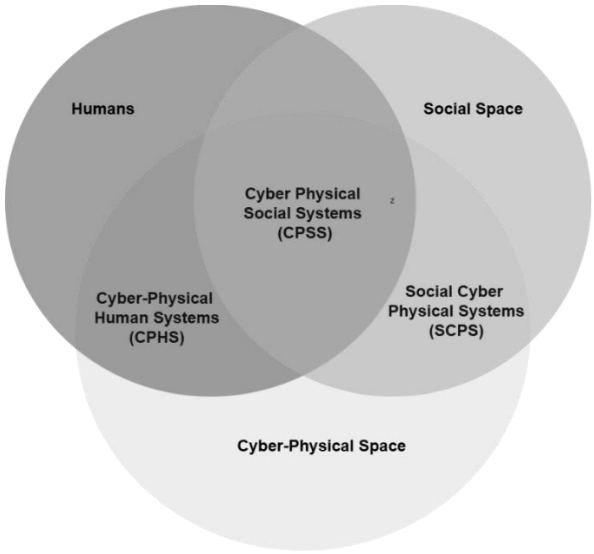


Figure 2.4: Distinction among CPSS, CPHS, and SCPS

Thus, evolution in a CPSS is of immense significance, dealing with the rapidly evolving contexts by bringing about the necessary changes in the system's behavior and structure. We subscribe to the notion of decision-based design [56], which posits that the principal role of a designer is to make decisions. In this context, decision-based design serves as a framework for managing the complexities of E-CPSS, thus guiding designers in making better informed decisions that leverage the inherent self-adaptive and self-organizing characteristics of these systems.

We define E-CPSS as systems that arise from continuous and interdependent interactions among the tri-space unlike standard adaptive systems which adapt within limited or constrained spaces, which distinguishes it

from general adaptive systems. While many adaptive systems rely on predefined rules of system design, E-CPSS incorporates human and social influence in its evolution as stakeholder needs and interests play an active role in defining system objectives, making the system more evolutionary and dynamically adaptable compared to classic adaptive models. Furthermore, traditional adaptive systems often function according to a clear optimization function and objective, whereas E-CPSS can shift objectives based on changing stakeholder needs.

E-CPSS shares several conceptual similarities with self-adaptive, self-evolving, and self-replicating systems, yet they also exhibit distinctive attributes, particularly due to the incorporation of social dynamics and decision-based design. Like self-adaptive systems, E-CPSS are designed to autonomously adjust their behavior and structure in response to changing environmental and stakeholder conditions. This aligns with their core characteristic of self-adaptability, enabling them to respond to dynamic social contexts and preferences. E-CPSS also demonstrates self-organizability, resembling self-evolving systems in its capacity to restructure internal architectures over discrete time states, attributes essential for navigating uncertain and complex environments. However, what sets E-CPSS apart is its integration of social entities as active agents in the system's evolution, influencing objectives and adaptation mechanisms through evolving cognitive, emotional, and behavioral inputs—something not typically modeled in conventional self-evolving systems. Compared to self-replicating systems, which emphasize autonomous reproduction or duplication of components (often inspired by biological systems), E-CPSS do not necessarily replicate themselves; rather they reorganize and reconfigure based on scenario recognition, context awareness, and emergent behavior driven by collective human interaction. Furthermore, while the vision of self-evolving computing systems [5] emphasizes an autonomous evolutionary engine for structural adaptation, E-CPSS embeds evolution into the triadic interaction of cyber, physical, and social spaces, making stakeholder feedback and social environment shifts integral to evolution rather than mere triggers. Thus, E-CPSS possess capabilities of self-adaptation, self-organization, and evolution akin to these intelligent systems, but they extend beyond them by embedding human-centric decision-making, social cognition, and multi-actor network effects as core design drivers, which are crucial for real-time and robust decision support in social-technical domains.

While we define the E-CPSS by incorporating knowledge about systems, evolution, and CPSS itself, the challenge remains for designers to understand and efficiently design systems by realizing the characteristics of the E-CPSS that help the system self-adapt and self-organize. These

characteristics are discussed in detail in the following section as a list of requirements to design E-CPSSs. The list of these characteristics stems from the review of the current literature and is shown in Table 2.3. In Table 2.3, we list the characteristics of an E-CPSS that we identified in Table 2.2, and we add additional characteristics that designers should consider when they design an E-CPSS. Where it helps the reader, we link each characteristic in Table 2.3 to the corresponding entry in Table 2.2. The evolving characteristics mentioned in Table 2.2 are the overlapping themes found common across the literature during our survey, which acts as a starting point to understand and realize the probable characteristics an E-CPSS must possess, post which the characteristics that are key to the performance of the E-CPSS are discussed in Table 2.3 in terms of their description and utility in the decision-making process. In Table 2.3, the rows represent the characteristics an E-CPSS must have, which we identified in Table 2.2. In contrast, the columns represent the description of each of these characteristics and their role in the decision-making process by the E-CPSS, thereby helping the designer design such systems. Through our study of the state of the art, we identified that a comprehensive review of the key characteristics of an E-CPSS is missing in the literature. These features are critical in realizing evolving capabilities of cyber-physical-social systems, enabling them to self-adapt and self-organize in response to changes in the external environment and cope with dynamic contexts. Further, as detailed in Table 2.3, they are critical. The system needs context awareness, which inherently allows designers to cater to the individual and emergent behaviors of the system. These characteristics are crucial not only for allowing adaptation to the environment but also for assisting in the decision-making process of the system. The characteristics shown in Table 2.3 are vital for systems designers to have a requirements list while designing an E-CPSS. These can be used to explain how the system can adapt, grow, and continue to function effectively over time. Organizations and institutions can meet evolving needs and maintain robust performance value by addressing these characteristics.

Table 2.3: Characteristics of an E-CPSS

Characteristic	**Description**	**Utility in Decision-making**

Scenario recognition	The process of identifying and understanding different scenarios or situations by analyzing data from various sources, including sensors, social media, and other information streams, to identify patterns and anomalies that indicate the occurrence of specific scenarios	Multiple possible scenarios and their consequences are evaluated, and the decision by an E-CPSS can be made, allowing the system to adapt to the changing scenario
Context Awareness	The ability of the system to interpret the changes in the surrounding environment and adapt to them. Realization of the correct context from the data and information sources is extremely important to interpret the needs and preferences of the stakeholders.	An E-CPSS is context-aware. It can identify the context in which the decision must be made. In an evolving environment, the context updates dynamically, allowing the system to evolve and adapt.
Organizability	The capability of the system to autonomously and collaboratively organize its components and resources to achieve desired system behavior or function.	System entities show structural and behavioral modification to organize themselves and perform functions to achieve the desired objectives.
Self-Adaptability	The ability of a system to adjust or modify its behavior, structure, or function autonomously in response to changing conditions or requirements to adapt to the changing scenario and context in a dynamic environment.	Systems having self-adaptability undergo a change in behavior and structure to adapt to the external environment.
Modularity	Breaking down the system into smaller components that can be developed independently, which can then be combined or interconnected to form a larger CPSS architecture.	Modularity enables system entities to undergo structural changes as part of the evolutionary process.

Scalability	The capability of a system to manage increased workloads, growing data volumes, or expanding requirements by scaling up or down while maintaining performance, functionality, and efficiency.	Interaction among the resources is managed based on interaction between the system and the dynamic environment. E-CPSS carefully expends or saves resources based on the function.
Interoperability	The capability of systems entities to communicate, exchange information, and cooperate effectively in a seamless manner by integration and interaction of heterogeneous technologies and subsystems allowing them to work together harmoniously to achieve common goals.	Interoperability makes interaction and information sharing among system elements possible for the system to actively adapt to the evolving environment without compromising on system performance and helping in real-time decision support
Robustness	The capability of a system to maintain stable and reliable performance even when subjected to variations, uncertainties, disturbances, and changes that occur as the system evolves over time.	An E-CPSS subjected to a complex environment must possess robustness to manage the risks and disruptions with the objective of minimizing the effect on the decision-making process.
Resilience	The capability of a system to effectively adapt, recover, and continue effective functioning of the system in the face of various challenges and disruptions that arise as the system evolves.	The ability to mitigate these external risks allows the system to recover and perform effectively, providing robust satisficing design solutions.

Emergent Behavior	Complex and unexpected system-level behavior arises due to the interactions and interdependencies among components within the system. These behaviors are not explicitly programmed or designed into the system but emerge as a consequence of the interactions and relationships among the components.	Unexpected emergent patterns can be harnessed for exploring new solutions and possibilities that might not be apparent through traditional, rule-based approaches.
Decentralized Decision Making	A decision-making approach in which individual components or subsystems are empowered to make independent decisions based on local information and objectives. Each component has a degree of autonomy to react to its environment and make choices that support the system's goals.	It takes into consideration the individual decisions and interactions of all the nodes in the cyber, physical, and social space, allowing for improved resource allocation and distributed data handling.

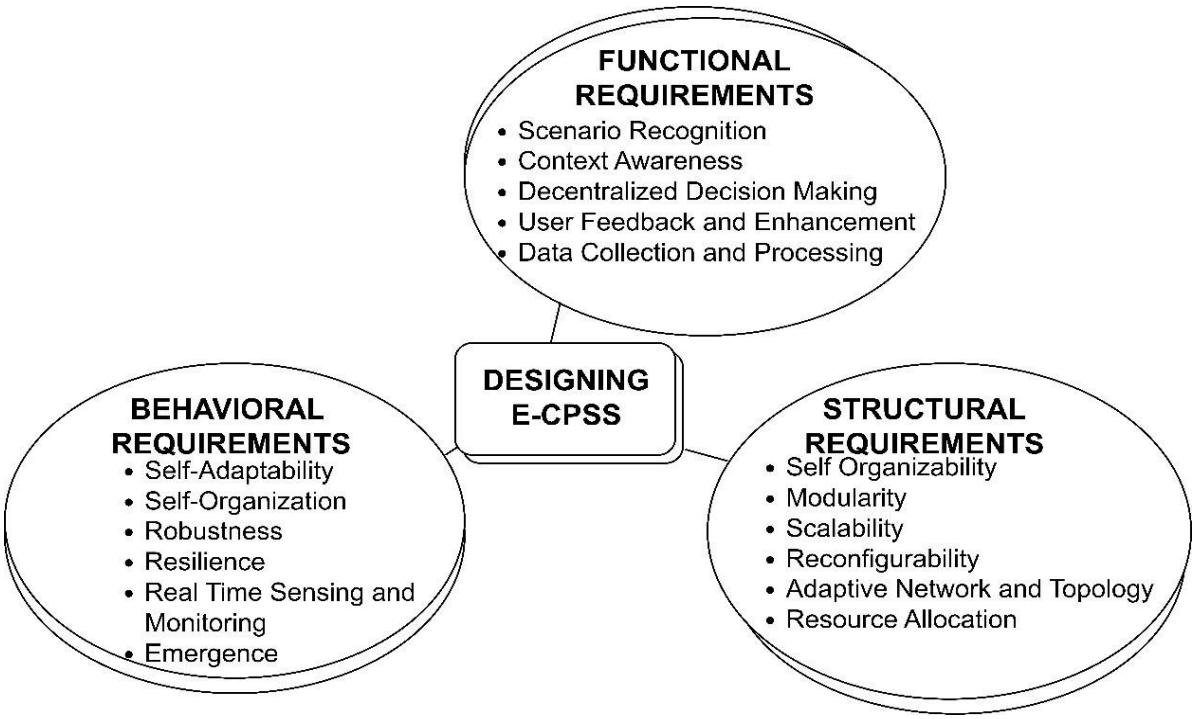


Figure 2.5: Classification of Design Requirements. Dependencies among Functional, Behavioral, and Structural Requirements illustrate how evolution in user feedback and preferences as well as changes in society in the cyber-physical-core expands the CPSS social envelope.

Frameworks that handle both cyber-physical integration and the dynamic, context-aware, and emergent behavior of the social dimension are necessary for designing an E-CPSS. Some models offer these features, which have been documented in the literature. By incorporating psychological models into AI-driven agents, Parallel Cognition is utilized to establish a connection between artificial and human cognitive processes, enabling systems to anticipate and recommend actions based on stakeholder behavior [22]. To manage the social complexity of E-CPSS, ACP (Artificial societies, Computational experiments, Parallel execution) employs agent-based modeling and dual-world simulations to simulate real-world social behaviors for ongoing assessment and management [76]. Additionally, we utilize Knowledge Automation with AI to mitigate noise and redundancy in cyber-physical-social data under uncertainty, thereby enhancing real-time decision-making [4]. Techniques such as Force Field Control and ADACOR enable self-organizability by allowing components to reorganize in response to stimuli, with applications from swarm robotics to adaptive manufacturing [8] [21]. Meanwhile, Generative Adversarial Networks (GANs) facilitate scenario-based learning and context refinement in human-centered environments [101]. The Hierarchical Bayesian Risk Graph Model (HRBGM) assesses risk and user behavior using Markov processes to predict disruptions, whereas cognitive models, such as MAPE-K, incorporate memory and knowledge management, enabling feedback loops crucial for real-time adaptation [110] [112]. By predicting system evolution through information dynamics, predictive models such as VAR (Vector Auto Regression) and GPR (Gaussian Process Regression) quantify node behavior and ensure system resilience [94, 95]. The 7C architecture enables knowledge synthesis and system evolution by introducing a layered framework that facilitates collective intelligence and self-organization [82] [113]. By translating high-level stakeholder needs into system-level design and capturing interdependencies among the cyber, physical, and social domains, MBSE (Model-Based Systems Engineering), and particularly MDSEA, can be used in structuring and simplifying E-CPSS design [2]. Together, these models form a comprehensive toolkit for designing E-CPSS that can not only self-adapt and self-organize but also evolve structurally and behaviorally in response to dynamic social contexts, ensuring resilient and context-aware system performance. These modeling frameworks align with the progression toward next-generation CPSs, where system behavior is no longer confined to managing energy, material, and information streams, but expands to cognitive, social, and value-driven domains [6]. In Section 2.1.10, there is a discussion of verification, validation practices that log state histories, assign trust scores to prior-

state evidence, and re-validate when objectives shift.

2.3.3 Behavioral and Structural Requirements

An E-CPSS undergoes behavioral and structural changes to adapt to its external environment and achieve the specific function for which it is designed. Connecting the characteristics of a system with its function, behavioral, and structural requirements for realizing these characteristics is essential for the successful design and development of such systems. The system's purpose can be established by categorizing the characteristics under specific functional requirements. Corresponding behavioral requirements can define how the system should adapt and respond to changing conditions to effectively exhibit those characteristics. Meanwhile, structural requirements emphasize the system's structural characteristics, enabling it to evolve and reconfigure itself as needed. Gero's Function-Behavior- Structure (FBS) framework [89] is a promising framework to study how an artifact's function, behavior, and structure interact.

Based on our study of the current literature, various models and frameworks exist that have the capability to realize evolving characteristics to ensure reliable system functionality. The survey of the literature and the selection of the model were done based on functional requirements identified through the realization of evolving characteristics that the system must possess to meet its intended design objective. Supplementary modeling details are provided in the Appendix. The fundamental design principle of these models can be further extended to design for an E-CPSS, complementing the realization of these characteristics in an E-CPSS design.

Based on this study we choose to build on the Function-Behavior-Structure approach recommended by Gero [89, 103]. The classification of the design requirements is done on the most recent and acceptable definition whose basis is formed by three classes of variables describing different aspects of a design object, which can be extended to complex systems such as E-CPSS [89]:

1 Function (F) variables: describe the theology of the object, that is, what it is for.
2 Behavior (B) variables describe the attributes that are derived or expected to be derived from the object's structure(S) variables, that is, what it does.
3 Structure (S) variables: describe the components of the object and their relationships, that is, what it is.

The FBS framework of the situated world and the proposed definition devised by Gero are used

to discuss and identify the system's key functional, behavioral, and structural requirements for realizing evolving attributes in an E-CPSS. This calls for an adaptive design approach wherein the system's capabilities are not restricted by rigid functionalities but allow it to continuously interpret, respond to, and reshape itself based on shifting stakeholder needs, environmental conditions, and internal state changes. We treat adaptation in the design of an E-CPSS as a fundamental design principle, ensuring that evolving characteristics are embedded from inception through intentional modeling of the system's responsiveness. In Figure 2.5, the functional, behavioral, and structural requirements in terms of the characteristics of an E-CPSS, which are to be considered by a designer to design an E-CPSS, are shown. These design requirements are fundamental in nature. The list of requirements can be further extended to achieve a specific system objective and include specific requirements like cognitive, social, economic, smartness, personalization, sustainability, etc. The significance of each characteristic and its alignment with the categorization is explained as follows.

1) Functional Requirements to Design E-CPSS

Functional requirements are the foundational requirements that a designer must consider while designing an E-CPSS, as these help establish and realize the purpose that the system would fulfill and is being designed for. It is also critical to examine the changes in the system characteristics with changes in the system states over time. These requirements guide the designer to ensure that the system's core functions align with the evolving goals and attributes it needs to exhibit over time. Scenario recognition is a critical functional requirement to ensure that the system recognizes the purpose for which it is designed. Combined with context awareness, scenario recognition enables a designed system to propose situationally appropriate decision options based on specific scenarios. Therefore, design rules for building frameworks to design for E-CPSS should be scenario-based [114]. Scenario-based self-organization systems are discussed in the state of the art with recognition of scenarios such as disaster relief situations [8] and home cleaning robots [73]. Incoming data from physical and human sensors can be analyzed to look for patterns and understand the context and scenario [81]. Context awareness, resulting in the realization of stakeholder needs, can be used to design for E-CPSS by understanding the purpose it is intended to serve, considering the needs of the social entities, and comprehending the system's intended function. A real-time complex event discovery platform for E-CPSS can be used to recognize and predict evolving scenarios [74]. Decentralized decision-making followed by social interactions is another critical functional

requirement a designer must consider realizing the system's true purpose. In an agent-based model, a certain degree of autonomy for individual components or agents allows the system to make autonomous decisions based on changing conditions to fulfill the system's purpose. Evolving systems such as ADACOR respond to internal and external changes by self-organizing and re-establishing system performance and functionality [21]. To provide realistic scenario information, the current and historical context information of entities is considered, which is collected, analyzed, and estimated to predict realistic evolving scenarios, allowing for decentralized decisions to be made [104]. The design of a context-aware E-CPSS involves utilizing interactions from social entities to narrow down, expand, or modify contextual factors, thereby improving scenario recognition and system performance in subsequent iterations of the decision-making process [101]. Interactions with social entities in E-CPSS provide essential contextual information depending upon the objective of the stakeholder to realize and predict a more precise function and purpose. The utility of social interaction in CPSS is shown in an airport situation using the Dynamic Social Structure of Things (DSSOT) [85]. These functional requirements are essential for defining the core functionality of the E-CPSS and ensuring that it can fulfill its intended purpose while adapting to changing needs and environmental advancement.

2) Behavioral Requirements to Design E-CPSS

Behavioral requirements are crucial to define the behavior of a system and are instrumental in realizing evolving characteristics, that is, attributes in a system. A designer should consider these requirements to design key attributes in the system to allow the system to undergo behavioral changes to evolve and satisfy the function realized. Behavioral requirements provide the "how" by specifying how the system should behave and how the designer wants it to behave in various situations and changing conditions. The behavioral requirements are critical from the evolution standpoint as they allow the designer to predict and comprehend how evolutionary behavior affects system performance in the face of environmental uncertainties and complexities. These requirements offer a roadmap for the system's dynamic behavior, guiding it in adjusting its actions, making real-time decisions, and continuously adapting its responses to meet evolving goals and address uncertainties.

The key requirements to realize evolutionary behavior in E-CPSS are self-adaptability and self-organizability. An E-CPSS subjected to an uncertain and complex environment must self-adapt and

self-organize to adapt to the changing environment, which involves evolving scenarios and contexts. Cyberspace and the network layer should support self-adaptation capabilities to ensure reliable and predictable communication between the components of the E-CPSS and between the E-CPSS and its environment. Physical devices should adapt their operations based on the results of context monitoring and their own self-assessment of their current state [84]. Social agents in the social space must exhibit self-organizability and self-adaptability in response to external disruptions and risks, thereby mitigating and managing those risks. Designing for self-adaptability in E-CPSS also ensures providing agility and responsiveness [86]. Adaptation is also discussed between humans and the CPS to reduce human cognitive load for decision-making, provide backup for malfunctions in CPS, and respond to external disruptions [90]. The ability of an E-CPSS to evolve can be attributed to its property to deal with risks and disruptions. To address this, an E-CPSS must possess and demonstrate robustness and resilience as its behavioral attributes. These act as key behavioral design requirements to ensure the safe and efficient working of the system. Approaches such as DSSOT (Dynamic social structure of Things) are discussed, which requires interaction with system entities and has a robust network to deal with heterogeneous data that can disrupt system functioning [85]. Resilience is utilized in power systems and smart grids by ensuring system capabilities to evaluate grid operation exposed to unexpected extreme disturbances and hazards, thereby leveraging the reliable and robust distribution of energy resources [7]. Agents in an E-CPSS collaborate and collectively work in a decision-making process, leading to emergent behavior in the system wherein the structure and behavior of the system are not what it is designed for; rather, something unplanned and unprogrammed appears. Coordination with autonomy in a decision-making process may lead to providing design solutions different from the intended objective. Collective decision-making, synchronization, and synergistic collaborations are important characteristics that should be considered as a behavioral requirement when designing an E-CPSS, as they enable the system to achieve goals that would be difficult to accomplish alone [108]. For instance, in a disaster response E-CPSS, emergency response teams might collaborate in unforeseen ways to manage a crisis effectively. In essence, behavioral requirements provide the flexibility and agility required for the system to evolve and maintain its performance in fulfilling its objective, ensuring that it can navigate the complexities of an ever-changing environment while consistently embodying the desired characteristics. These requirements are due to the attributes that are derived or expected to be derived from the structural

changes exhibited in an E-CPSS due to its evolutionary behavior over time. Through the layered adaptation model (micro-meso-macro) as explained in Section 2.2.3, we support the mapping of system behaviors and responses across scales, aiding designers in identifying which adaptations may accumulate into evolutionary changes.

3) Structural Requirements to Design E-CPSS

Structural requirements are indispensable because they shape its ability to adapt and evolve. These requirements will help the E-CPSS realize the structural changes it must undergo to evolve and fulfill its intended function. These requirements define the components and their relationships to support the system's evolving attributes effectively. Structural requirements dictate how the system can be realized to expand, modify, or reconfigure without causing disruptions, fostering its scalability and resilience in the face of uncertainties. Furthermore, they seamlessly integrate new technologies and components, allowing the system to stay current and adaptable in an ever-changing environment.

The key structural requirement of designing for evolutionary behavior and addressing the changing relationships between the components in an E-CPSS is self-organization that assumes the role of creating and maintaining a logical network structure on top of a dynamically changing physical structure topology wherein autonomous and dynamic structuring of components, context information, and resources takes place [73]. Structural self-organizing mechanisms must be considered a key requirement to enable the system to deal with a dynamic environment. Self-organization at various system levels allows for achieving evolutionary reconfigurable and modular systems, such as in an ADACOR system [21]. Reconfigurability changing external needs, enabling the system to dynamically reconfigure and rearrange its resources accordingly. On the other hand, modularity in complex systems enables systems to add, remove, replace, or reconfigure modules and resources in response to changing needs. Using the key requirement of designing for modular and reconfigurable characteristics, a designer can ensure the system portrays structurally evolving attributes in the system architecture. Evolvable Assembly Systems (EAS) are defined as self-adaptive reconfigurable production systems that showcase modularity and reconfigurability wherein individual modules can cooperate with other modules, creating new skills enhancing interoperability and emergent behavior to cope with the constantly changing market demands and

aim to promote efficient and economical production systems [86, 100]. Adding or removing modules from the system architecture enables the system to demonstrate scalability in handling the uncertain magnitude of data and uncertainties that interact with it [85]. We treat scalability as a key structural requirement when we design an E-CPSS. We design for the handling of large, heterogeneous data and for the delivery of solutions while we maintain computational tractability. Generative systems [75] and decentralized systems show scalable characteristics due to ubiquitous computing and connectivity [107]. Interoperability is another key structural requirement, wherein entities constantly interact with one another and undergo structural changes collectively and in synchronization to adapt to an external changing environment, thereby accomplishing system goals and objectives. Interoperability among network enterprises is a crucial requirement for maintaining sustainable network systems and architecture in the face of emerging challenges arising from the evolution of enterprises and new information systems [2]. Adaptability in E-CPSS for resource and task allocation is a requirement designers must consider, leveraging human cognitive abilities and responding to external factors by enabling synchronization between various entities of the E-CPSS. In essence, structural requirements serve as the foundation upon which evolving characteristics are built, ensuring that the system's structure remains versatile and capable of fulfilling its evolving goals and attributes. A well-designed framework with the essential structural requirements allows the system to respond to changing user needs and seamlessly integrate new technologies, ensuring the E-CPSS evolves coherently and effectively over time.

2.4 E-CPSS Cognitive Map for Decision-Making

Based on the preceding information and the study of the literature, we believe that three assumptions are essential for modeling an E-CPSS.

Assumption 1: Tri-space co-evolution

The cyber, physical, and social artifacts must each have adaptation hooks and be co-balanced over the system life cycle. This requires life cycle modeling tools that retain “state history” for all three spaces.

Assumption 2: There must be an Explicit cognition layer

Human reliability assessment and collective cognition must be modelled as state-aware, updatable constructs, not heuristics. This establishes the need for work on cognition metrics, trust calibration,

and real-time cognitive state inference.

Assumption 3: Decision-centric design

Every functional, behavioral, or structural choice should be traceable to a quantifiable decision policy that remains revisable as social contexts drift. Encourages "decision provenance" logging and adaptive policy engines.

Consider a system or conceptualization of a system meeting [these] assumptions. The tri-space (cyber, physical, and social spaces) is individually unique but interacts with the spaces of the others. As time progresses, the tri-spaces interact and are observed through the cognition layer. After an arbitrarily long time at t_0 a policy decision is made that impacts all spaces, another arbitrarily long t_1 continues until punctuated by another policy decision, *ad infinitum*. All changes within each timestep are traceable to the prior policy decision(s). During each time step, actors within and among the tri-spaces interact and re-evaluate their conditions.

As an abstract example, consider a resource usage policy on groups A, B, and C. A policy decision following t_0 redirects a portion of a resource from group A to group B while maintaining the total flow of the resource. As t_1 moves along, an intelligent cyber infrastructure places more weight on flow uncertainty reported by physical sensors at B to assess resource flow and how social actors interact in ways that benefit each group --- perhaps A and C momentarily align goals in opposition to B. In this case, cyber adaptation has occurred as the intelligent system re-evaluates the stochastic weights for resource monitoring to most effectively monitor the system, and social adaptation has occurred through A and C to address resource shortages. If information is shared between all stakeholders, a designer assesses the system state at the end of t_1, verifying the occurrence of each interaction. Shared observations allow B to note the alignment of A and C through explicit cognition, allowing internal correction. If the system state during t_1 was not satisfactory, this analysis can be used to inform a policy change leading to t_2. Since all parties have access to the cognition layer, they have a modicum of agency over the consequences of their actions within the timestep, and they trust that the system may operate with autonomy, as each stakeholder will have a voice in the next policy step.

To model an E-CPSS coherently, researchers should describe and predict collective performance across behavioral contexts and capture the aspects along which tasks, strategies, and network structures emerge. Based on the state-of-the-art techniques identified in our literature review, we

summarize and consolidate design tools for E-CPSS. We discuss selected techniques for modeling evolution in a CPSS and for explaining the decision-making process in an E-CPSS. The sampling of techniques we present here is selected based on a systemized form of the review process wherein emphasis is laid on modeling techniques used to model and showcase evolving characteristics, as mentioned in Table 2.2.

We illustrate the evolution and decision-making process in an uncertain and dynamic environment, recognizing the importance of context awareness. The cognitive map serves as a structured mental representation that is created to organize and navigate understanding of physical spaces, environments, concepts, or information. It is a mental framework to store, process, and recall spatial or conceptual information to make sense of surroundings and make decisions. The cognitive map represents an E-CPSS with evolving characteristics in the tri-space, facilitating real-time information sharing among the three spaces. The various processes and workings of the model can be explained using the example of a water utility system, wherein a change in the environment external to the system brings about a change in the user requirements, causing the system to evolve. The interconnections of physical assets (pumps, pipes, and valves), processes, and social interactions (customers, community engagement, policymakers) with the emergence of digital technology adoption (software, data, and network adaptation) are transforming water utilities into CPSSs for collective decision-making process which is why it is chosen as an example to demonstrate the working of the cognitive map [88]. When designers understand the interconnections among these components, they can enhance services, leading to the design of more robust systems and the provision of design solutions in application areas such as water demand management, customer service, and the detection of contamination incidents. After analyzing how cyber, physical, and social spaces interact as discussed in Section 2.1.2, as well as the evolving characteristics outlined in Table 2.3, and the requirements discussed in Section 2.3.3, we developed the cognitive map shown in Figure 2.6. In Subsection 2.4.1, we describe the features of the cognitive map. In Subsection 2.4.2, we explain how designers use the cognitive map.

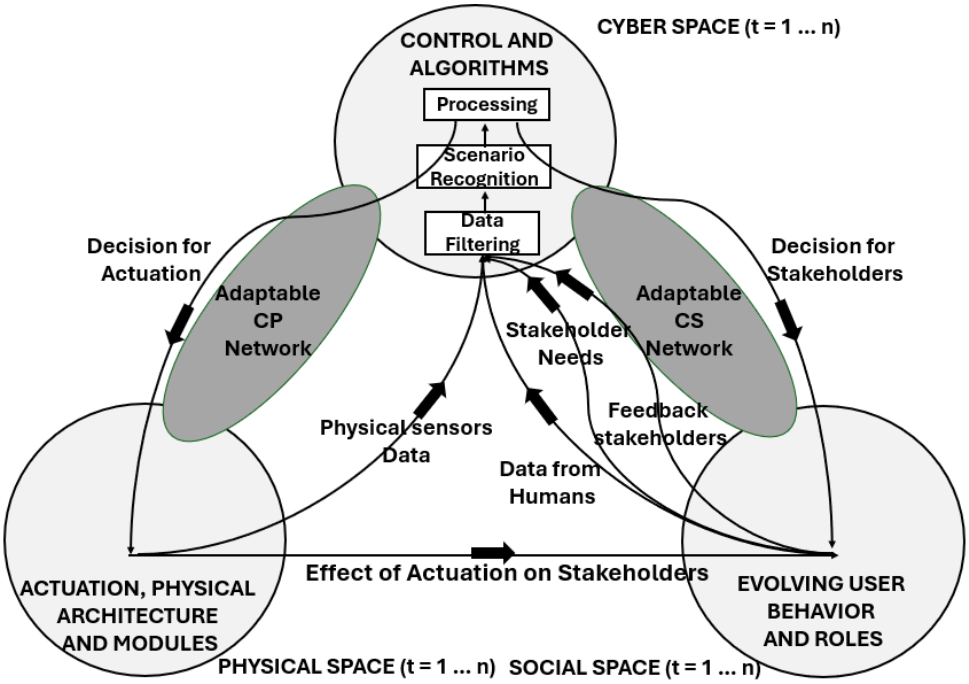


Figure 2.6: Cognitive Map for the E-CPSS

2.4.1 Features of the Cognitive Map

We use Figure 2.6 to show the cyber, physical, and social spaces and illustrate the conceptual linkages between each space. Each space briefly includes information about the action that occurs there, and interactions flow between them. In cyberspace, heterogeneous data from humans and physical sensors is homogenized, and patterns and similarities are observed in the homogenized data to recognize scenarios and context and understand the requirements. Decision makers interpret the context and make a decision based on that interpretation. Physical space encompasses the actuating capabilities and sensors for sensing and data generation. Components are modular in that, with changes in stakeholder demands, the physical configuration can be rearranged, removed, or added to satisfy the requirements; that is, the space undergoes structural evolution to achieve the updated state. The social space consists of humans and their interdependencies, social networks, and social analytics. Humans' emotions, roles, and behaviors are a part of the social space. The three spaces act as spatiotemporal zones, being both space- and time-dependent, and exhibit changes in structure and behavior over time. As the environment changes over time, so do the spaces. The evolving behavior of the system is observed when the system function updates due to changing

context and recognizes updated scenarios. The network for sharing information between the Cyber- and Physical- space (CP) and Social- and Cyber-space (CS) also evolves due to the dynamic sharing of large amounts of heterogeneous information between the various spaces. The shift in user/stakeholder preferences in the social space, as well as the interactions triggered by unsatisfactory decisions, is responsible for the change in context. Thus, the CPSS also evolves to counter this dynamic change caused by the changes in its social space in real-time by self-adapting to suit the needs of the stakeholders and providing satisfactory solutions. It is important to note that cognition is inherently embedded within and across these spaces, though not explicitly labeled as a separate "cognitive space" in Figure 2.6. Cognition in this framework can be understood in two complementary ways. First, human cognition, which is central to stakeholder behavior and is thus represented within the Social-Space, where evolving user needs and feedback loops are captured. Second, system-level cognition refers to the system's computational ability to interpret, learn from, and adapt to changing stakeholder demands and environmental contexts over time. This is within Cyber-Space. These functions collectively enable the system to reason about its own performance and evolution, effectively embodying cognitive capabilities, thus representing cognition as not a separate fourth space in the cognitive map but as a cross-cutting dimension.

To explain the cognitive map with a hypothetical example, we consider a rural water utility system showing capabilities and characteristics of an E-CPSS as discussed in the chapter. Unlike water facilities in urban areas, which have extensive resources and facilities, rural utility systems have a lower tolerance to uncertainty without impacting the customer. A rural utility system with robust framework to reduce the risk of disruptions in the system's functioning. Limited resources necessitate a properly designed, robust framework to reduce the risk of disruptions. To explain the cognitive map, we have assumed that the stakeholders' initial requirements were to have "water supply access" for purposes such as drinking and cleaning. The authorities understood the requirements well, and the system worked in a coordinated manner to ensure that every household had "water supply access." Based on the interaction with social entities, necessary changes were made until every requirement was fulfilled to achieve a steady state. However, the stakeholders' requirements begin to change in response to environmental changes. Climate change, urban activities, and differences in lifestyle result in water stagnation, and the stakeholders' requirements are updated to "public health protection." This change in requirements results in corresponding change in decisions by the E-CPSS in collaboration with policymakers and allocates the required

resources to meet the changing demand. Hence, with the change in requirement (the driver), the system's function, behavior, and structure (driven) change. The mechanism of adapting to this change is discussed through the cognitive map. This cognitive map builds upon the adaptation-evolution concept described earlier in Section 2.2.3, where dynamic feedback loops at micro and meso levels may lead to macro-level transformations over time.

2.4.2 Workings of the Cognitive Map

Initially, we gather heterogeneous data, including multiple preferences and physical parameters from physical and social sensors, and we send it to Cyberspace. For example, in a water utility system, data from social sensors includes stakeholders' requirements, such as clean water and regular supply, and data from physical sensors includes the water's pH level, the water's temperature, etc. In Cyberspace, heterogeneous data is analyzed and processed to create homogeneous data using computational methods that utilize the necessary data based on stakeholder requirements, while filtering out the rest to ease computational complexity. The filtered-out data is stored in the servers' databases, which can be accessed for other decision-making situations and requirements. Computations take place on homogeneous data, and scenario recognition takes place. For example, the requirement of having "water supply access" is recognized. Once the scenario is recognized, context awareness is observed in Cyberspace with its evolved cognitive capabilities. The decision for physical actuation and operation is sent to the physical space based on the context in which actuation takes place.

Modularity is observed here as a change in requirement brings about repositioning, removal, replacement, or addition of structural modules. These transitions are natural point at which to perform verification and validation. As an example, electromechanical solenoid valves receive signals to maintain the flow and pressure of the water supply, ensuring uninterrupted access until new signals with updated commands and controls are received. The decision can also be sent directly to the social entities if the requirement does not require physical actuation [Cyber-Social Interaction (CSS)]. For example, a query about the duration and timing of water usage during the day is provided directly by Cyberspace. The effect of actuation in physical space is observed in Social space, regarding human behavior, cognition, and emotion [Physical-Social Interaction (PSS)]. Suppose the decision taken by the system is unsatisfactory. In that case, social entities generate feedback through social interaction. It is sent to Cyberspace as input simultaneously with

the existing heterogeneous data incoming from physical and social sensors, allowing for new and updated scenario recognition and context awareness, thus showcasing an evolving attribute. For example, the supply of water is not available to every stakeholder or potential user, and a query is sent. Within the social Space, user roles and behavior evolve to the different levels of preferences by direct and indirect stakeholders and changes in the environment. Thus, humans act as users of the system. The changing preferences and objectives of the user make the data dynamic as the information also evolves, and subsequently, the situation and context get updated. For example, climate change and urban activities have degraded the water quality level, and stakeholders are now concerned about the users' health. Thus, the system's objective updates from ensuring water supply access to providing a clean water supply. The decision-making process is repeated until the final objective of the users/stakeholders is satisfied and the system achieves a steady state. Thus, the evolving nature of the CPSS can be observed from the above description of the cognitive map's operation. With the changes in the users' and stakeholders' requirements, the context updates, which in turn bring about changes in the tri space of the system. The changes continue to occur until the requirements are met; hence, the system achieves a steady state. This shift illustrates macro-evolution within the system: multiple localized (micro) and operational (meso) adaptations accumulate, prompting a functional transition consistent with the adaptation-evolution ideation provided in Section 2.2.3. The utility of the cognitive map is to design and understand the complex interactions within systems such as E- CPSS. In designing an E-CPSS, a cognitive map is valuable for understanding interdependencies, identifying key components and relationships, and interpreting the characteristics of such systems. Designers can use such cognitive maps to support decision-making, whether it involves modeling social dynamics, incorporating user preferences, or identifying potential risks and vulnerabilities within the system. In summary, the proposed cognitive map offers a comprehensive and dynamic representation of an E-CPSS, supporting the design, analysis, and management of complex systems. This representation can be leveraged to ensure that system components work synergistically to satisfy stakeholder needs and achieve system goals. An important avenue for future work involves addressing situations where the preferences or opinions of social entities diverge, conflict, or are even antagonistic. While the current cognitive map outlines an evolving decision-making process driven by feedback from stakeholders, it assumes a level of alignment, whereas in practice, multi-stakeholder environments often present value conflicts or strategic opposition. Future extensions of the proposed framework can be

explored to incorporate conflicting decision strategies that explicitly account for competing interests which will enhance the system's robustness in managing social diversity, enabling E-CPSS to function effectively under complex decision scenarios.

2.5 Key Findings in Designing E-CPSS

In this section, we highlight our findings and provide suggestions, shedding light on key insights for advancing design engineering in the E-CPSS. The key contribution of this chapter is to provide a pathway for system designers to design an E-CPSS effectively from the perspective of enhancing the decision-making process. This contribution is achieved by addressing the various entities of the system and the relationships among these entities, as discussed in the cyber, physical, and social spaces of an E-CPSS, in Section 2.1.2. From our discussion in Section 2.1.2 and Table 2.2, it is clear that the Social space is integral to the decision-making process. To effectively design and tackle an E-CPSS, a list of requirements is formulated in Section 2.3.3, encompassing perspectives that cater to the system's function, behavior, and structure. We use the evolving characteristics in Table 2.3 to guide the development of foundational frameworks, approaches, and methods by leveraging existing tools to address E- CPSS design.

Finally, various features of the models and frameworks from Figure 2.6, the requirements list from Section 2.3.3, and the characteristics from Table 2.3 are combined to present a cognitive map of an E-CPSS, as shown in Figure 2.6. The cognitive map presented will be of utmost value in the field of design engineering, helping designers understand the workings of an E-CPSS during the decision-making process when subjected to an uncertain environment while considering the interests of all the stakeholders involved. Based on our findings, the challenges need to be addressed to advance design engineering for E-CPSS.

Challenge 1: Addressing evolving Cyber-Physical-Social systems.

E-CPSS are designed to possess immense computational, physical, and cognitive capabilities to deal with external changes when there are uncertainties and complexities. Hence, addressing the evolving attributes of such systems, along with frameworks, approaches, and methods to model these characteristics, becomes critical. The challenges associated with addressing E-CPSS include data security and privacy, dynamic resource management, human-centric design, and ethical and societal implications.

Challenge 2: *Uniform modeling of cyber, physical, and social spaces.*

A state-of-the-art review presents different techniques, methods, and frameworks for modeling the cyber, physical, and social spaces in different scenarios and assumptions. However, the literature still faces the challenge of uniform modeling of the tri-space that can be used to model the functioning and behavior of a CPSS in every possible scenario. Challenges such as data heterogeneity, computational complexity, dynamic nature, and interdisciplinary integration are second to the need for uniform modeling of cyber, physical, and social spaces.

Challenge 3: *Realizing the social dimension and incorporating endogenous and exogenous social space to design E-CPSS effectively.*

The role of social dimension and interactions bringing about evolving characteristics in a CPSS is relatively untapped in the state of the art due to for the challenge of modeling and predicting emotions and behavior associated with the social dimension. The social space has both endogenous and exogenous characteristics that need to be recognized to address the social interactions resulting from the system's evolving behavior, and to understand how these interactions give rise to evolutionary behavior within the system. Balancing the impact of internal social dynamics (endogenous) with external societal factors (exogenous) and understanding how they interact and influence each other is a challenge that needs to be addressed to incorporate these seamlessly into the design of E-CPSS.

Potential Avenues for Exploration

The potential avenues for exploration are presented as investigative questions to foster the enhancement of Design Engineering in E-CPSS, subsequently enhancing the research in this field. These potential avenues for exploration are presented from the standpoint of three categories: fundamental knowledge for E-CPSS, Technology and Computation, and Design Engineering. We propose these questions for further exploration from the perspective of attaining synergistic

creation of new knowledge in the three categories mentioned above:

- How can we model public acceptance for design options for social problems, such as an E-CPSS?
- What is the method for modeling human judgment (and other human reliability assessment factors) to enhance the social capabilities for decision support for an E-CPSS?
- How can we enhance real-time decision-making by enhancing the predictive capabilities while designing an E-CPSS?
- What are the computational capabilities required to establish scenario-based modeling techniques for addressing the social space in an E-CPSS?
- What foundational knowledge is necessary for predictive analysis, which is critical for E-CPSS and enables the anticipation of different states within an E-CPSS by examining past states and forecasting future states?
- What new knowledge is required to propose the FBS design model for an E-CPSS, which will account for the evolving system behavior and structure for the desired function?
- What mathematics is required to address E-CPSS and subsequently design it for the decision support of complex systems?
- How will trust, cooperation, coordination, etc., affect the effective design of a robust E-CPSS? How can these characteristics be quantified and incorporated for E-CPSS design?

Further, the computational research opportunities in designing Evolving Cyber-Physical-Social Systems are detailed in [3].

2.6 Closing Remarks for Chapter 2

In this chapter, we present our findings on the dynamic nature of an E-CPSS, which is influenced by the inclusion of social space involving human behavior, emotions, and cognition, thereby differentiating it from a CPS. The importance of the social domain in the decision-making process is discussed, and the cause-and-effect relationship between evolution in social space and changes in the overall system architecture is presented in Section 2.2. After a critical review of the state of the art, we identify key research questions, as discussed in Section 2.3, that must be addressed to

accommodate the evolving nature of a CPSS in the decision-making process. These questions highlight challenges in the research of E-CPSS, ranging from addressing the social dimension to identifying key evolving attributes and approaches for modeling these attributes in the design of an E-CPSS. To address these research questions, we describe an E-CPSS in Section 2.3.2, detailing the system's evolving features and the behavioral and structural changes the system undergoes due to this evolving nature. Various key features emerging in the system due to its evolutionary behavior, such as self-organization, self-adaptability, resilience, interoperability, context-awareness, and many more, are presented in Table 2.3 to aid designers in designing an E-CPSS. We infer that modeling and framework resources for designing for E-CPSS are areas of untapped potential. Therefore, potential models and frameworks are summarized in Figure 2.6, which are commonly used in numerous applications, such as smart cities, intelligent transportation, and water utility systems. For this, Table 2.2 is presented to identify and classify our study of the state of the art in order to categorize and summarize our findings comprehensively. To realize evolving attributes, various requirements are inferred from the standpoint of Gero's FBS framework, as shown in Figure 2.5, to contribute to the design research community with the concept of concurrent designing for such systems, thus underlining the critical functional, behavioral, and structural requirements to realize and showcase evolving characteristics discussed in Section 2.3.3. We also present a cognitive map of an E-CPSS, as shown in Figure 2.6, which concerns water utility systems and explains the decision-making process in an uncertain and complex environment with constantly changing stakeholder needs and preferences. Researchers and designers can use the map to understand the decision-making mechanism in an evolving environment, which will help them expand or revamp the model according to their objectives and problem statements. We present the cognitive map is to guide further research into the development of design procedures for developing frameworks for E-CPSS.

As a CPSS evolves, new security and privacy challenges emerge. Studying system evolution helps identify potential vulnerabilities and design measures to ensure the security and privacy of sensitive data and operations in a large, adaptable network that handles a large volume of heterogeneous data. An in-depth study of E-CPSS can foster innovation by identifying opportunities for new features, services, or applications that arise as the system evolves. As CPSS evolves, models can be updated to reflect new requirements, technologies, and insights, allowing for continuous improvement and adaptation to changing needs. Emergent behavior commonly

observed in Collective Adaptive Systems (CAS) is a distinctive and important characteristic of an E-CPSS that cannot be predicted, controlled, or accounted for. Future studies can expand to explore modeling and computational techniques for controlling and predicting emerging attributes, thereby preventing misbehavior and unaccounted, unplanned risks that disrupt system performance.

Considering the complex nature of E-CPSS and the transformative implications for system design, it is necessary not only to summarize the key takeaways but also to engage in a deeper, critical reflection. In what follows, we articulate the propositions, discuss deficiencies and limitations of the work, and identify near-future research opportunities grounded in existing gaps and interdisciplinary needs.

We invite other researchers to test our framework using longitudinal case studies. We ask researchers to design studies that cover the physical, cyber, cognitive, social, and human domains and to report findings that refine the tri-space + cognition paradigm. We base our call for longitudinal case studies on established methods across multiple domains. Lee (2008) documents hardware and software integration cycles in automotive CPS engineering [115]. In those projects, engineers adjust sensor networks and control algorithms in response to field data. Autrey and co-authors (2018) describe a self-assessment rubric. Researchers apply that rubric to four successive device prototypes [116]. The researchers adopt hardware layouts and control logic after each iteration of the prototype. Baxter and Sommerville (2011) describe co-design workshops. Participants refine user interfaces and organizational workflows in each workshop [117]. Bradshaw and co-authors (2011) explain five iterations of a service-design challenge. Teaching teams shift their risk-taking behaviors and collaboration patterns, demonstrating cognitive and social adaptation [117]. We invite researchers to use these methods to test our tri-space plus cognition framework through longitudinal studies that span the physical, cyber, cognitive, social, and human domains.

To encourage these longitudinal studies, we outline the scholarly propositions we propose and discuss some potential next steps.

2.6.1 Scholarly Propositions: Synthesizing Key Contributions

Building on the assumptions described in Section 2.1, we propose the following propositions as anchor points for the theoretical and practical advancement of E-CPSS:

Proposition 1: Tri-space coevolution must be designed explicitly.

E-CPSS are not merely integrations of cyber, physical, and social systems - they are coevolutionary systems where changes in one space continuously influence and are influenced by the others. This requires systemic modelling across time and interaction layers.

Proposition 2: Social space is not an add-on but a primary design dimension.

The social entity, comprising human cognition, emotion, roles, and social context, should be modelled as an important design object, not as an external influence or end-user. This reframing fundamentally shifts the architecture and purpose of the system.

Proposition 3: Evolutionary capability must be an inherent attribute of system architecture.

Self-adaptability, self-organizability, resilience, and modularity are not optional enhancements; they are design-time imperatives for systems intended to survive and perform in dynamic, uncertain environments.

Proposition 4: Cognitive modelling and decision-based design must converge.

Decision-making in E-CPSS must be situated within context-aware cognitive frameworks that incorporate scenario recognition, trust calibration, and adaptive policy management.

Proposition 5: Transdisciplinary integration is essential.

No single discipline can capture the complexity of E-CPSS. The design, analysis, and governance of these systems require collaboration across engineering, cognitive science, social science, ethics, AI, and policy.

2.6.2 Limitations and Deficiencies

While in this chapter we lay a critical theoretical foundation, the following are some of the limitations that we acknowledge and present as opportunities for other researchers in transdisciplinary domain-

A. Conceptual and theoretical orientation

The cognitive map, modelling frameworks, and characteristics discussed remain conceptual. Additional work is needed for the empirical validation or simulation-based proof of concept. To evaluate the proposed E-CPSS cognitive map, some of the conceptual strategies that can be used include simulation-based testing, scenario-driven prototyping and using metrics and benchmarks. Simulation-based testing includes agent-based simulation to model and validate emergent behavior in E-CPSSs.

B. Limited operationalization of cognitive features

Although cognitive layers are discussed, the mechanisms for modelling human cognition, decision fatigue, preference evolution, and trust dynamics remain underdeveloped.

C. Absence of implementation frameworks

There is no complete reference architecture or development environment that operationalizes E-CPSS holistically across tri-space interaction and stakeholder alignment.

D. Ethical, legal, and social implications (ELSI) are treated insufficiently.

This work does not yet embed data governance, fairness, transparency, or equity into the design logic. These issues are paramount when systems act autonomously in social contexts. There is an immense opportunity for researchers to work in the quantification and management of these aspects.

E. Computational complexity and scalability concerns

As system state and stakeholder diversity grow, the computational load of real-time adaptation and scenario modelling remains a major barrier to practical deployment.

These acknowledged gaps motivate the next section on research opportunities.

F. Practical Constraints in Implementing E-CPSS

Although the E-CPSS paradigm proposed is conceptual in nature, its deployment faces several practical constraints, including human-generated biases, inconsistencies, privacy, and trust issues when integrating sensitive data. Furthermore, technical limitations and interoperability make E-CPSS implementation challenging, Regulatory and ethical considerations must also be addressed. Finally, user adoption and behavioral resistance will always be a challenge to ensure a smooth transition of the E-CPSS from theory to practice.

2.6.3 Near-Future Research Opportunities

To mature E-CPSS theory and enable real-world deployments, several avenues of inquiry are

proposed:

A. Theoretical and Mathematical Modelling

- Develop mathematical formulations to represent evolutionary feedback loops across cyber, physical, and social spaces.
- Extend graph theory, network science, and dynamical systems theory to model adaptive, multi-agent interactions.

B. Cognitive Systems Integration

- Construct computational models of cognition that account for stakeholder trust, decision inertia, emotion, and preference evolution.
- Explore the use of reinforcement learning, attention networks, and meta-learning to simulate human-like adaptation.

C. Design Frameworks and Toolkits

- Build design toolkits that translate FBS requirements into modular software-hardware architectures.
- Develop simulation environments that combine agent-based models, digital twins, and real-time sensor data.

D. Ethical and Human-Centered Design

- Incorporate principles from value-sensitive design, algorithmic transparency, and privacy-preserving computation.
- Metrics for social fairness, inclusion, and legitimacy can be defined in ways that permit thresholds for mitigation or retraining. When such thresholds are documented in a design dossier, external reviewers may use them to accredit the process after state transitions.
- Integrating human input is essential for the evolution of the system; however, human-generated data might contain bias that can further propagate through the system. Therefore, what possible mechanisms should be considered in the early design stages to detect and mitigate these biases is a potential avenue for further exploration. Promising future research in this challenge includes studies on bias detection and monitoring, adaptive mitigation strategies, and continuous feedback as early design considerations while designing for an E-CPSS..

E. Coupled Evolution of CPS and CPSS

- As self-improvement in cyber-physical systems reshapes stakeholder behavior, this, in turn,

guides the next wave of self-improvement. Find methods that capture and quantify the bidirectional influence across the cyber, physical, and social domains. To address this challenge, some possible ways of addressing it are:

- Build multi-layer simulations to link learning agents with agent-based social modules
- Conduct longitudinal field studies to measure how real stakeholders respond to adaptive features
- Integrate formal verification methods with civic engagement workshops to ensure that mathematical guarantees and social legitimacy advance together.

G. Systems of Systems and Emergent Complexity

- Examine how adaptation and evolution unfold in systems of systems, where interdependence and emergent properties go beyond single-system boundaries.
- Develop modelling frameworks to capture cross-system feedback loops and unintended interactions in evolving socio-technical ecosystems.

H. Mission-Critical and Safety-Critical Applications

- Explore adaptation and evolution in mission-critical domains such as healthcare, transportation, and defense, where reliability and safety must be guaranteed even under change.
- Identify mechanisms that allow evolution without compromising predictability or introducing unacceptable risk to critical operations.

I. Failure Management and Security Assurance

- Advance methods for failure detection, containment, and recovery in evolving systems, ensuring that breakdowns become opportunities for resilience rather than points of collapse.
- Investigate how adaptation and evolution mechanisms themselves may introduce new vulnerabilities, and design strategies for security assurance that evolve alongside the system.

J. Ethical and Normative Dimensions

- Study how adaptation and evolution affect fairness, accountability, and transparency in evolving socio-technical systems.
- Develop frameworks to ensure that evolutionary trajectories remain aligned with human values and societal legitimacy, particularly in contexts of contested interests.
- **Interoperability and Standardization**
- Explore how adaptation and evolution can be managed across heterogeneous systems, organizations, and jurisdictions.

- Develop common standards and interoperability mechanisms that allow evolving systems to interact reliably without undermining trust or security.

2.6.4 Final Reflections: Toward a Cognitively Enriched Design Epistemology

The future of E-CPSS is not just a technical evolution. It is a cognitive, social, and ethical transformation. We argue that systems must evolve from being merely connected to becoming cognitively aware, socially embedded, and ethically grounded. This evolution will be catalyzed by developments in artificial intelligence (AI) and cognitive system modelling, which are poised to redefine how systems interact with humans and make decisions.

- **AI and Cognitive Directions for E-CPSS**

  E-CPSS design must harness emerging capabilities from AI and human cognition modelling, including:
- **Neuro-symbolic AI** for hybrid reasoning using structured rules and behavioral patterns.
- **Multi-agent reinforcement learning (MARL)** for decentralized, real-time adaptation in heterogeneous systems.
- **Affective computing and sentiment analysis** to measure stakeholder reactions, emotional trust, and social acceptance.
- **Cognitive digital twins** that simulate not just physical entities but also human behavior, enabling predictive and reflective planning.
- **Explainable AI (XAI)** to ensure that decisions made in evolving contexts remain interpretable, justifiable, and trustworthy.
- **Trust-aware learning architectures**, where system behavior evolves based on dynamic interpersonal trust assessments.

These developments are essential for ensuring that E-CPSS can function autonomously yet remain aligned with human needs and expectations.

From Reactive to Reflective Systems

Future E-CPSS must become reflective systems that both react to and actively assess the implications of their decisions, evaluating feedback, and adjusting their goals. This requires architectures that support meta-cognition, scenario-based forecasting, and cross-domain learning. Such systems should be capable of:

- Predicting emergent risks based on weak signals from the social space.
- Revising goals in response to collective feedback loops.
- Learning from unsuccessful decisions and adjusting design strategies over time.

By embedding cognition, social understanding, and adaptive design logic into the very fabric of engineered systems, we move toward a future where our technologies evolve with us, not just for us.

We advocate for a design epistemology that:

- Places human cognition and trust at the center of system architecture,
- Employs AI not just for efficiency but for ethical reasoning and stakeholder alignment,
- Supports continuous co-evolution with the environments and communities it serves.

The future of system design is no longer about building smarter machines. It is about designing intelligent, ethical, and socially aware partners in decision-making. Through this chapter, we set the stage for that future.

References

[1] Abera, Y. B., Naudet, Y., and Panetto, H., 2020, "A New Paradigm and Meta-Model for Cyber-Physical-Social Systems," *IFAC-Papers-Online*, 53(2), pp. 10949–10954.

[2] Agostinho, C., Ducq, Y., Zacharewicz, G., Sarraipa, J., Lampathaki, F., Poler, R., and Goncalves, R. J., 2016, "Towards a Sustainable Interoperability in Networked Enterprise Information Systems: Trends of Knowledge and Model-Driven Technology," *Computers in Industry*, 79, pp. 64–76.

[3] Allen, J. K., Nellippallil, A. B., Ming, Z., Milisavljevic-Syed, J., and Mistree, F., 2023, "Designing Evolving Cyber- Physical-Social Systems: Computational Research Opportunities," *ASME Journal of Computing and Information Science in Engineering*, 23(6), p. 060815.

[4] Zhang, J. J., Wang, F. Y., Wang, X., Xiong, G., Zhu, F., Lv, Y., Hou, J., Han, S., Yuan, Y., Lu, Q., and Lee, Y., 2018, "Cyber- Physical-Social Systems: The State of the Art and Perspectives," *IEEE Transactions on Computational Social Systems*, 5(3), pp. 829–840.

[5] Weyns, D., Bäck, T., Vidal, R., Yao, X., and Belbachir, A. N., 2023, "The Vision of Self-evolving Computing Systems," *Journal of Integrated Design and Process Science*, 26(3–4), pp. 351–367.

[6] Horváth, I., 2023, "Designing Next-Generation Cyber-Physical Systems: Why is it an Issue? ," *Journal of Integrated Design and Process Science*, 26(3–4), pp. 317–349.

[7] Arghandeh, R., von Meier, A., Mehrmanesh, L., and Mili, L., 2016, "On the Definition of Cyber-Physical Resilience in Power Systems," *Renewable and Sustainable Energy Reviews*, 58(May), pp. 1060–1069.

[8] Zhou, Z., Sun, Y., Ouyang, C., Gan, Z., and Ming, Z., 2022, "A Design Framework for Evolving Cyber-Physical-Social System (CPSS) Based on Force Field," *ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers Digital Collection.

[9] Smirnov, A., Levashova, T., Shilov, N., and K., S., 2014, "Ontology for Cyber-Physical-Social Systems Self-organization," *Conference of Open Innovations Association FRUCT*, pp. 101–107.

[10] Horvath, I., 2024, "Extended Editorial: Research to Support Cognitive Engineering of Intellectualized Cyber-Physical-Social-Human Systems," *Journal of Integrated Design and Process Science*, 27(3–4), pp. 153–168.

[11] Wang, F. Y., 2010, "The Emergence of Intelligent Enterprises: From CPS to CPSS," *IEEE Intelligent Systems Journal*, 25(4), pp. 85–88.

[12] Liu, Y., Peng, Y., Wang, B., Yao, S., and Liu, Z., 2017, "Review on Cyber-Physical Systems," *IEEE/CAA Journal of Automatica Sinica*, 4(1), pp. 27–40.

[13] Hasan, M. K., Habib, A. A., Shukur, Z., Ibrahim, F., Islam, S., and Razzaque, M. A., 2023, "Review on Cyber-physical and Cyber-security System in Smart Grid: Standards, Protocols, Constraints, and Recommendations," *Journal of Network and Computer Applications*, 209, p. 103540.

[14] Pivoto, D. G. S., de Almeida, L. F. F., Righi, R., and Rodrigues, J. P. C., 2021, "Cyber-physical Systems Architectures for Industrial Internet of Things Applications in Industry

4.0: A Literature Review," *Journal of Manufacturing Systems*, 58 Part A.(January 2021), pp. 176–192.

[15] Zhou, J., Chung, C. Y., Wang, H., and Liu, N., 2021, "Multi-microgrid Energy Management Systems: Architecture, Communication, and Scheduling Strategies," *Journal of Modern Power Systems and Clean Energy*, 9(3), pp. 463–476.

[16] Amiri, Z., Heidari, A., Navimipour, N. J., Unal, M., and Mousavi, A., 2024, "Adventures in Data Analysis: A Systematic Review of Deep Learning Techniques for Pattern Recognition in Cyber-Physical-Social Systems," *Multimedia Tools and Applications*, 83, pp. 22909–22973.

[17] Leng, J., Wang, D., Shen, W., Li, X., Liu, Q., and Chen, X., 2021, "Digital Twins-based Smart Manufacturing System Design in Industry 4.0: A Review," *Journal of Manufacturing Systems*, 60(July), pp. 119–137.

[18] Wang, X., Wang, L. T., Xie, X., Jin, J., and Deen, M. J., 2017, "A Cloud-edge Computing Framework for Cyber-Physical-Social Services," *IEEE Communications Magazine*, 55(11), pp. 80–85.

[19] Glasson, B. C., 1989, "Model of System Evolution," *Information and Software Technology*, 31(7), pp. 351–356.

[20] Baldauf, M., Dustdar, S., and Rosenberg, F., 2007, "A Survey on Context-aware Systems," *Journal of Ad Hoc and Ubiquitous Computing*, 2(4), p. 263.

[21] Barbosa, J., Leitão, P., Adam, E., and Trentesaux, D., 2015, "Dynamic Self-organization in Holonic Multi-agent Manufacturing Systems: The ADACOR Evolution," *Computers in Industry*, 66, pp. 99–111.

[22] Ye, P., Wang, X., Zheng, W., Wei, Q., and Wang, F.-Y., 2022, "Parallel Cognition: Hybrid Intelligence for Human-Machine Interaction and Management," *Frontiers of Information Technology and Electronic Engineering*, 23(12), pp. 1765–1779.

[23] Bucchiarone, A., D'Angelo, M., Pianini, D., Cabri, G., De Sanctis, M., Viroli, M. C., R., and Dobson, S., 2020, "On the Social Implications of Collective Adaptive Systems," *IEEE Technology and Society Magazine*, 39(3), pp. 36–49.

[24] Gilb, T., 1985, "Evolutionary Delivery versus the ‘Waterfall Model’," *SIGSOFT Software Engineering Notes*, 10(3), pp. 49–61.

[25] Kluver, J., 2000, The Dynamics and Evolution of Social Systems: New Foundations of a Mathematical Sociology, v. 29, Springer Science and Media.

[26] Partridge, L. D., 1982, "The Good Enough Calculi of Evolving Control Systems: Evolution Is Not Engineering," *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 242(3), pp. R173–177.

[27] Lee, J. D., and See, K. A., 2004, "Trust in Automation: Designing for Appropriate Reliance," *Human Factors*, 46(1), pp. 50–80.

[28] Hoff, K. A., and Bashir, M., 2015, "Trust in Automation: Integrating Empirical Evidence of Factors that Influence Trust," *Human Factors*, 57(3), pp. 407–434.

[29] Carayannis, E. G., and Morawska-Jancelewicz, J., 2022, "The Futures of Europe: Society 5.0 and Industry 5.0 as Driving Forces of Future Universities," *Journal of the Knowledge Economy*, 13, pp. 3445–3471.

[30] Kephart, J. O., and Chess, D. M., 2003, "The Vision of Autonomic Computing," *Computer*, 36(1), pp. 41–50.

[31] Madni, A. M., and Madni, C. C., 2018, "Architectural Framework for Exploring Adaptive Human Machine Teaming Options in Simulated Dynamic Environments," *Systems*, 6(4), p. 44.

[32] Woods, D. D., 2006, "Essential Characteristics of Resilience," *Resilience Engineering: Concepts and Precepts*, E. Hollnagel, D. D. Woods, and N. Levenson, eds., Ashgate.

[33] Zhou, Y., Yu, F. R., Chen, J., and Kuo, Y., 2019, "Cyber-Physical-Social Systems: A State-of-the-Art Survey, Challenges and Opportunities," *IEEE Communications Surveys and Tutorials*, 22(1), pp. 389–425.

[34] Grieves, M., and Vickers, J., 2017, "Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," *Transdisciplinary Perspectives on Complex Systems*, F.-J. Kahlen, S. Flumerfelt, and A. Alves, eds., Springer, pp. 85–113.

[35] Sornette, D., Davis, A. B., Ide, K., Vixie, K. R., Pisarenko, V., and Kamm, J. R., 2007, "Algorithm for Model Validation: Theory and Applications," *Proceedings of the National Academy of Sciences*, 104(16), pp. 6562–6567.

[36] Anton, A. I., and Potts, C., 1998, "The Use of Goals to Surface Requirements for Evolving Systems," *20th International Conference on Software Engineering*, pp. 157–166.

[37] Hahn, H. A., 2013, "The Conundrum of Verification and Validation of Social Science-Based Models. ," *Conference on Systems Engineering Research*, pp. 878–887, 16, January.

[38] Yilmaz, L., 2006, "Validation and Verification of Social Processes within Agent-Based Computational Organization Models," *Computational and Mathematical Organization Theory*, 12(4), pp. 283–312.

[39] Belfrage, M., Johansson, E., Lorig, F., and Davidson, P., 2024, "[In]Credible Models – Verification, Validation and Accreditation of Agent-Based Models to Support Policy-Making," *Journal of Artificial Societies and Social Simulation*, 27(4).

[40] Alizon, F., Shooter, S., and Simpaon, T. W., 2006, "Reuse of Manufacturing Knowledge to Facilitate Platform-Based Product Realization," *Journal of Computing and Information Science in Engineering*, 6(2), pp. 170–178.

[41] Harland, P. E., and Yorur, H., 2015, "Decisions in Product Platform Development Projects," *International Journal of Innovation and Technology Management*, 12(1), p. 1550001.

[42] Simpson, T. W., Maier, J. R. A., and Mistree, F., 2001, "Product Platform Design: Method and Application," *Research in Engineering Design*, 13, pp. 2–22.

[43] Pasandideh, S., Pereira, P., and Gomes, L., 2022, "Cyber-Physical-Social Systems: Taxonomy, Challenges, and Opportunities," *IEEE Access*, 10, pp. 42404–42419.

[44] Jirgl, M., Bradac, Z., and Fiedler, P., 2018, "Human-in-the-loop Issue in Context of the Cyber-Physical Systems," *IFAC- PapersOnLine*, 51(6), pp. 225–320.

[45] Schirner, G., Erdogmus, D., Chowdhury, K., and Padir, T., 2013, "The Future of Human-in-the-loop Cyber-Physical Systems," *Computer*, 46(1), pp. 36–45.

[46] Tyworth, M., Giacobe, N., Mancuso, V., McNeese, M., and Hall, D., 2013, "A Human-in-the-Loop Approach to Understanding Situation Awareness in Cyber Defence Analysis," *EAI Endorsed Transactions on Security and Safety*, 13(May), pp. 1–10.

[47] Holland, J. H., 1992, Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT Press, Cambridge, MA.

[48] Etang, A., Fielding, D., and Knowles, S., 2011, "Does Trust Extend Beyond the Village? Experimental Trust and Social Distance in Cameroon.," *Experimental Economics*, 14, pp. 15–35.

[49] Gharib, M., Lollini, P., and Bondavalli, A., 2017, "Towards an Approach for Analyzing Trust in Cyber-Physical-Social Systems," *12th System of Systems Engineering Conference (SoSE)*.

[50] Chopra, K., and Wallace, W., 2003, "Trust in Electronic Environments," *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference. IEEE*, p. 10.

[51] Huybrechts, B., Nelson, D., Nelson, T., O'Shea, N., and Dufays, F., 2024, "Conform and Oppose through Numbers: Quantifying Hybrid Organizations at the International Cooperative Alliance," *Journal of Management Studies*, 61(4), pp. 1393–1426.

[52] Nunez, A. N., Giachetti, R. E., and Boria, G., 2009, "Quantifying Coordination Work As a Function of the Task Uncertainty and Interdependence," *Journal of Enterprise Information Management*, 22(3), pp. 1741–0398.

[53] Marcus, M., and Le, H., 2013, "Interactive Effects of Levels of Individualism—Collectivism on Cooperation: A Meta-analysis," *Journal of Organizational Behavior*, 34(6), pp. 813–834.

[54] Turner, J. H., 1988, A Theory of Social Interaction, Stanford University Press, Palo Alto, CA.

[55] Weber, M., 1968, Economy and Society: An Outline of Interpretative Sociology (Vol. 1), Bedminster Press, New York, NY.

[56] Mistree, F., Smith, W., Bras, B., Allen, J. K., and Muster, D., 1990, "Decision-Based Design: A Contemporary Paradigm for Ship Design," *Transactions- Society of Naval Architects and Marine Engineers*, 98, pp. 565–597.

[57] Guo, L., Allen, J. K., and Mistree, F., 2025 (In Press), Architecting a Satisficing Strategy for Desinging Complex Systems, Springer, Cham, Switzerland.

[58] Panchal, J., Fernández, M. G., Paredis, C. J. J., Allen, J. K., and Mistree, F., 2008, "A Modular Decision-Centric Approach for Reusable Design Processes," *Concurrent Engineering: Research and Applications*, 15(3), pp. 309–323.

[59] Messer, M., Panchal, J. H., Allen, J. K., and Mistree, F., 2011, "Model Refinement Decisions Using the Process Performance Indicator," *Engineering Optimization*, 43(2), pp. 741–762.

[60] Ming, Z., Wang, G., Yan, Y., Panchal, J. H., Allen, J. K., and Mistree, F., 2018, "An Ontology-Based Representation of Design Decision Hierarchies," *ASME Journal of Computing and Information Science in Engineering*, 18(1), p. 011001.

[61] Ming, Z., Sharma, G., Allen, J. K., and Mistree, F., 2019, "Template-Based Configuration and Execution of Decision Workflows in Design of Complex Engineered Systems," *Advanced Engineering Informatics*, 42, p. 100985.

[62] Baby, M., Guptan, A., Broussard, J., Allen, J. K., Mistree, F., and Nellippallil, A. B., 2024, "A Decision Support Network for Robust Multilevel Co-Design Exploration of Manufacturing Supply Networks," *ASME Journal of Mechanical Design*, 146(1), p. 111704.

[63] Mistree, F., Hughes, O. F., and Bras, B. A., 1993, "The Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm," *Structural Optimization: Status and Promise*, M. P. Kamat, ed., AIAA, Washington, D. C., pp. 247–286.

[64] Wang, G., Shang, X., Yan, Y., Allen, J. K., and Mistree, F., 2018, "A Tree-Based Decision Method for the Configuration Design of Reconfigurable Machine Tools," *Journal of Manufacturing Systems*, 149, pp. 143–169.

[65] Nellippallil, A. B., Rangaraj, V., Gautham, B. P., Singh, A. K., Allen, J. K., and Mistree, F., 2018, "An Inverse, Decision-Based Design Method for Integrated Design Exploration of

Materials, Products and Manufacturing Processes," *ASME Journal of Mechanical Design*, 140(11), p. 11403.

[66] Guo, L., Zamanisabzi, H., Neeson, T. M., Allen, J. K., and Mistree, F., 2019, "Managing Conflicting Water Resource Goals and Uncertainties in a Dam-Network by Exploring the Solution Space," *ASME Journal of Mechanical Design*, 141(3), p. 031702.

[67] Guo, L., Chen, S., Allen, J. K., and Mistree, F., 2020, "A Framework for Designing the Customer Order Decoupling Point to Facilitate Mass Customization," *ASME Journal of Mechanical Design*, 143(2), p. 022002.

[68] Samadiani, E., Joshi, Y., Allen, J. K., and Mistree, F., 2010, "Adaptable Robust Design of Multi-scale Convective Systems Applied to Energy Efficient Data Centers," *Numerical Heat Transfer, Part A: Applications*, 57(2), p. 121002.

[69] Sabeghi, M., Shukla, R., Allen, J. K., and Mistree, F., 2016, "Solution Space Exploration of the Process Design for Continuous Casting of Steel," *ASME Design Automation Conference*, ASME, pp. Paper Number DETC2016-59399.

[70] Guo, L., Allen, J. K., and Mistree, F., 2024, "Optimize or Satisfice in Engineering Design?," *Research in Engineering Design*, 35, pp. 239–267.

[71] Xue, X., Zhou, D., Wang, X., and Wang, F. W., 2023, "Computational Experiments for Complex Social Systems—Part III: The Docking of Domain Models," *IEEE Transactions on Computational Social Systems*, 11(2), pp. 1766–1780.

[72] Li, B., Cao, D., Tang, S., Zhang, T., Dong, H., Wang, Y., and Wang, F. Y., 2023, "Sharing Traffic Priorities via Cyber– Physical–Social Intelligence: A Lane-Free Autonomous Intersection Management Method in Metaverse," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(4), pp. 2025–2036.

[73] Smirnov, A., Kashevnik, A., and Ponomarev, A., 2015, "Multi-level Self-organization in Cyber-Physical-Social Systems: Smart Home Cleaning Scenario," *Procedia CIRP*, 30, pp. 329–334.

[74] Dao, M.-S., Pongpaichet, S., Jalali, L., Kim, K., Jain, R., and Zettsu, K., 2014, "A Real-time Complex Event Discovery Platform for Cyber-Physical-Social Systems," *Proceedings*

of the International Conference on Multimedia Retrieval (ICMR '14), Association for Computing Machinery, pp. 201–208.

[75] Lombardi, M., and Vannuccini, S., 2019, "Understanding Emerging Patterns and Dynamics Through the Lenses of the Cyber-Physical Universe," *Patterns*, 3(11), p. 10061.

[76] Wang, F.-Y., 2007, "Toward a Paradigm Shift in Social Computing: The ACP Approach," *IEEE Intelligent Systems Journal*, 22(October), pp. 65–67.

[77] Bennaceur, A., Ghezzi, C., Tei, K., Kehrer, T., Weyns, D., Calinescu, R., Dustdar, S., and Hu, Z., 2019, "Modelling and Analysing Resilient Cyber-Physical Systems," *2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pp. 70–76.

[78] Assunta, C., Guido, G., Silvestro, V., and Giusy, V., 2017, "Man-CPS Interaction: An Experimental Assessment of the Human Behavior Evolution," *2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry*, pp. 1–6.

[79] Dustdar, S., and Scekic, O., 2018, "On Managing the Social Components in a Smart City," *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 115-1163.

[80] Yilma, B. A., Panetto, H., and Naudet, Y., 2021, "Systemic Formalisation of Cyber-Physical-Social System (CPSS): A Systematic Literature Review," *Computers in Industry*, 129(August), p. 103458.

[81] Singh, V., Gao, M., and Jain, R., 2012, "Situation Recognition: An Evolving Problem for Heterogeneous Dynamic Big Multimedia Data," *Proceedings of the 20th ACM International Conference on Multimedia*.

[82] Yin, D., Ming, X., and Zhang, X., 2020, "Understanding Data-Driven Cyber-Physical-Social System (D-CPSS) Using a 7C Framework in Social Manufacturing Context," *Sensors*, 20(18), pp. Article number 18.

[83] Neves, P., and Onori, M., 2012, "Evolvable Production Systems: Approach Towards Modern Production Systems," *Proceedings of the 6th CIRP-Sponsored International Conference on Digital Enterprise Technology, Advances in Intelligent and Soft Computing*, G. Q. Huang, K. L. Mak, and P. G. Maropoulos, eds., Springer.

[84] Zeadally, S., Sanislav, T., and Mois, G. D., 2019, "Self-adaptation Techniques in Cyber-Physical Systems (CPSS)," *IEEE Access*, 7, pp. 171126–171139.

[85] Hussein, D., Park, S., Han, S., and Crespi, N., 2015, "Dynamic Social Structure of Things: A Contextual Approach in CPSS," *IEEE Internet Computing*, 19(3), pp. 12–20.

[86] Leitão, P., and Barbosa, J., 2016, "Dynamic Switching Mechanism to Support Self-organization in ADACOR Holonic Control System," *IFAC-Papers-Online*, 49(31), pp. 161–166.

[87] Colabinchi, S., Costantino, F., Di Gravio, G., Nonino, F., and Patriarca, R., 2021, "Discussing Resilience in the Context of Cyber Physical Systems," *Computers in Industry*, 160, p. 197534.

[88] Bhandari, P., Creighton, D., Gong, J., Boyle, C., and Law, K. M. Y., 2023, "Evolution of Cyber-Physical-Human Water Systems: Challenges and Gaps," *Technological Forecasting and Social Change*, 191(June), p. 122540.

[89] Gero, J., and Kannengiesser, U., 2004, "The Situated Function-Behaviour-Structure Framework," *Design Studies*, 25, pp. 373–391.

[90] Madni, A., Sievers, M., and Madni, C., 2018, "Adaptive Cyber-Physical-Human Systems: Exploiting Cognitive Modeling and Machine Learning in the Control Loop," *INSIGHT*, 21, pp. 87–93.

[91] Dorst, K., and Vermaas, P. E., 2005, "John Gero's Function-Behaviour-Structure Model of Designing: A Critical Analysis," *Research in Engineering Design*, 16(1–2), pp. 17–25.

[92] Lee, Y. T., 1999, "An Overview of Information Modeling for Manufacturing Systems Integration," NIST.

[93] Tu, Y. L., and Xie, S. Q., 2001, "An Information Modelling Framework to Support Intelligent Concurrent Design and Manufacturing of Sheet Metal Parts," *International Journal of Advanced Manufacturing Technology*, 18, pp. 873–883.

[94] Wang, Y., 2021, "Probabilistic Modeling of Information Dynamics in Networked Cyber-Physical-Social Systems," *IEEE Internet of Things Journal*, 8(19), pp. 14934–14947.

[95] Wang, Y., 2021, "Topology-informed Information Dynamics Modeling in Cyber–Physical–Social System Networks," *AI EDAM*, 35(3), pp. 316–331.

[96] Bagozi, A., Bianchini, D., and Antonellis, V. D., 2021, "Context-based Resilience in Cyber-Physical Production System," *Data Science and Engineering*, 6(4), pp. 434–454.

[97] Li, S., Zhao, S., Yuan, Y., Sun, Q., and Zhang, K., 2018, "Dynamic Security Risk Evaluation via Hybrid Bayesian Risk Graph in Cyber-Physical Social Systems," *IEEE Transactions on Computational Social Systems*, 5(4), pp. 1133–1141.

[98] van Beek, A., 2022, "A Decision-Centric Perspective on Evolving Cyber-Physical-Social Systems: Effectiveness, Group Value, and Opportunities," *ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers Digital Collection.

[99] Ye, P., and Wang, F.-Y., 2022, "Parallel Population and Parallel Human—A Cyber-Physical Social Approach," *IEEE Intelligent Systems Journal*, 37(5), pp. 19–27.

[100] Sanderson, D., Chaplin, J. C., and Ratchev, S., 2019, "A Function-Behaviour-Structure Design Methodology for Adaptive Production Systems," *The International Journal of Advanced Manufacturing Technology*, 105(9), pp. 3731–3742.

[101] Mukhopadhyay, S., Liu, Q., Collier, E., Zhu, Y., Gudishala, R., Chokwitthaya, C., DiBiano, R., Nabijiang, A., Saeidi, S., Sidhanta, S., and Ganguly, A., 2020, "Context-aware Design of Cyber-Physical Human Systems (CPHS) " *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*.

[102] Liu, Z., Yang, D., Wen, D., Zhang, W., and Mao, W., 2011, "Cyber-Physical-Social Systems for Command and Control," *IEEE Intelligent Systems Journal*, 26(4), pp. 92–96.

[103] Gero, J., and Kannengiesser, U., 2003, "Function-Behaviour-Structure: A Model for Social Situated Agents," *Workshop on Cognitive Modeling of Agents and Multi-Agent Interactions, International Joint Conference on Artificial Intelligence*.

[104] Chavhan, S., Gupta, D., Chandana, B. N., Khanna, A., and Rodrigues, J. J., 2020, "IOT-based Context-aware Intelligent Public Transport System in a Metropolitan Area," *IEEE Internet of Things Journal*, 7(7), pp. 6023–6034.

[105] Xiong, G., Zhu, F., Liu, X., Dong, X., Huang, W., Chen, S., and Zhao, K., 2015, "Cyber-Physical-Social System in Intelligent Transportation," *IEEE/CAA Journal of Automatica Sinica*, 2(3), pp. 320–333.

[106] Tyszberowicz, S., and Faitelson, D., 2020, "Emergence in Cyber-Physical Systems: Potential and Risk," *Frontiers of Information Technology & Electronic Engineering*, 21(11), pp. 1554–1566.

[107] Leon, F., 2012, "Emergent Behaviors in Social Networks of Adaptive Agents," *Mathematical Problems in Engineering*, 2012, pp. 1–19.

[108] Li, Z., Sim, C., and Low, M. H., 2006, "A Survey of Emergent Behavior and its Impacts in Agent-based Systems," *IEEE International Conference on Industrial Informatics, 2006*, pp. 1295–1300.

[109] Estrada-Jimenez, L. A., Pulikottil, T., Peres, R. S., Nikghadam-Hojjati, S., and Barata, J., 2021, "Complexity Theory and Self-organization in Cyber-Physical Production Systems," *Procedia CIRP*, 104, pp. 1831–1836.

[110] D'Angelo, M. D., Caporuscio, M., and Napolitano, A., 2017, "Model-driven Engineering of Decentralized Control in Cyber-Physical Systems," *IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W)*.

[111] Horvath, I., and Abraham, G., 2025, "Transdisciplinary Shifts in System Paradigm-Driven Disciplines: Mechatronics as an Example," *Transdisciplinary Journal of Engineering and Science*, 16, pp. 177–205.

[112] Galesic, M., Barkoczi, D., Berdahl, A. M., Biro, D., Carbone, G., Giannoccaro, I., Goldstone, R. L., Gonzalez, C., Kandler, A., Kao, A. B., Kendal, R., Kline, M., Lee, E., Massari, G. F., Mesoudi, A., Olsson, H., Pescetelli, N., Sloman, S. L., Smaldino, P. E., and Stein, D. L., 2023, "Beyond Collective Intelligence: Collective Adaptation," *Journal of the Royal Society Interface*, 20, p. 20220736.

[113] Cabrera, O., Oriol, M., Franch, X., and Marco, J., 2021, "A Context-aware Monitoring Architecture for Supporting System Adaptation and Reconfiguration," *Computing*, 103, pp. 1621–1655.

[114] Rao, V., Joshi, A., Kang, S. M., Lin, S., Song, J., Miller, D., and Lambert, K. G. A., A., 2022, "Designing Privacy Risk Frameworks for Evolving Cyber-Physical Social Systems: Knowledge Gaps Illuminated by the Case of Autonomous Vehicles and Bystander Privacy," *ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers Digital Collection.

[115] Lee, E. A., 2008, "Cyber Physical Systems: Design Challenges," *11th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, pp. 363–369.

[116] Autrey, J. L., Sieber, J., Siddique, Z., and Mistree, F., 2018, "Leveraging Self-assessment to Encourage Learning Through Reflection on Doing," *International Journal of Engineering Education*, 4(2B), pp. 708–722.

[117] Baxter, G., and Sommerville, I., 2011, "Socio-technical Systems: From Design Methods to Systems Engineering," *Interacting with Computers*, 23(1), pp. 4–17.

Appendix for Chapter 2

Supplementary Modeling Details

Given the interdisciplinary nature of E-CPSS systems, a variety of modeling paradigms, frameworks, methods, and techniques were considered for designing an E-CPSS, as discussed in Section 1. This supplement elaborates on the theoretical underpinnings and characteristics of such approaches to provide a deeper and foundational understanding beyond the scope of the main discussion.

ACP Approach

The ACP Approach consists of: a) modeling Artificial societies by utilizing agent-based simulations involving human behaviors; b) analysis with Computational experiments, that is, finding an effective way to conduct experiments for system development; c) control through Parallel execution by running one or more artificial systems in parallel with a real system [76]. Behaviors emulated by artificial systems can be used to optimize for real system performance in real time [11].

ADACOR

In ADACOR, an adaptive production control approach to adaptively balance static and transient states in normal and unexpected conditions, is presented. Self-organization at the micro (behavioral self-organization) and the macro level (structural self-organization) allows the system to respond to internal and external changes over time and re-establish system performance and functionality [21]. In the event of uncertainty, holons self-organize, providing agility and responsiveness to the emergency by transitioning from a transient state to a stationary state as soon as possible.

Collective Adaptation Framework

This framework is used to manage the behavior of agents using different integration strategies and network architectures in an unpredictable environment [112]. Researchers use the framework to evaluate whether designers have enabled the system to behave coherently and adaptively and to modify its control rules in response to external or internal stimuli [23].

.

Force Field Control Method

Researchers commonly use this method when they equip swarm robots with self-organizing capabilities. It consists of two parts: the Inner and Task Force Fields. Inner Force Fields enable robots to maintain operating distance with proper directionality. The Task Force Field determines the group behavior to perform the system's specific task while interacting with the environment. The combination of the two force fields allows the system to self-organize while continuously interacting with the environment to achieve system objectives [8].

GAN (Genetic Adversarial Networks)

GAN are used to generate mixture models, which allow appropriate fusion of existing designs and knowledge of human interactions as obtained from IVE (Immersive Virtual Experiments) for modeling design characteristics. IVE experiments are conducted based on physical experiments, and the feedback is used for modifying the set of specific influential contextual factors that should be considered to improve system design and performance [101].

HRBGM (Hierarchical Bayesian Risk Graph Model)

A hidden Markov model (HMM) is introduced to model the dynamic user activities [97]. The HMM and the BRG, if used to identify influences of activities of neighboring nodes, to analyze potential risks caused by dynamic user activities [112].

Knowledge Automation

This method is used to perform computation on cyber-physical and social data while taking into consideration various uncertainties, redundancies, and inconsistencies in the system [4].

MAPE-K

The importance of cognition, as discussed in the Parallel Cognition framework, led to an evolved version of MAPE known as MAPE-K (Monitor-Analyze-Plan-Execute-Knowledge) [110]. Knowledge is useful for modeling self-adaptive systems to implement system functionalities, maintaining a steady and reliable flow of relevant information to other components and system nodes, and implementing the adaptation through the Monitor, Analyze, Plan, and Execute components. Cognition is developed in the form of constructive memory, and the system becomes capable of adapting, adjusting, and computing changing needs in real-time based on historical data.

MBSE (Model-Based Systems Engineering)

It can be used to model complex dynamic relationships and dependencies among system entities, facilitating the reduction of enterprise complexity. It utilizes principles to guide the transformation from

requirements/objectives to a detailed specification of the system's components' functions and features, including Human, IT, and Physical, which correspond to the social, cyber, and physical spaces of the E-CPSS. The promising framework addresses system dynamics that arise due to interoperability among enterprises [2].

Parallel Cognition

Parallel Cognition is composed of the artificial, and the real human cognitive system interacting with each other in three stages: descriptive, predictive, and prescriptive cognition. Social entity needs are realized, and computations are performed in the artificial world that allows the system to predict possible behavior to address those needs [22].

VAR and GPR

These are used to model information dynamics and design for fluctuating decision-making and the predicting capability of the nodes in a network. These models can be used to predict the evolution of node behavior, each having sensing and predictive capabilities quantified with a prediction probability. Thus, VAR and GPR can be used to model for interoperability and ensure resilience by predicting and defining allocation of resources to the various system nodes [94, 95].

The 7C Architecture

Among the seven levels of this framework, collective intelligence is involved in inducing self-adaptive capability. Inputs in the data to the information level, cyber level, and cognition level can also imbibe self-organization by analyzing and learning from user preferences [82]. Whereas, the knowledge level and the "Knowledge-to-wisdom" levels are the ones where the system evolution might take place due to the dynamic cognitive capability of the components [109].

CHAPTER 3

OBSERVING E-CPSS: CPSS-STRUCTURED PREDICTIVE MODELING FOR SOCIOTECHNICAL SYSTEM DESIGN

Background and Context

In Chapter 2 we establish the theoretical ground: we define the design object through the tri-space coevolution model, we state through the eleven E-CPSS characteristics what any framework must achieve, and we justify our questions through the seven research gaps we identify. Of the five functional requirements that we establish in Chapter 2 for E-CPSS design, two come before all others: Scenario Recognition and Context Awareness. A designer who cannot read the current state of the system cannot structure decisions within it, cannot derive adaptive pathways across its transitions, and cannot validate any intervention against what is actually happening. Every subsequent capability in the DSIM depends on having a principled, design-relevant characterization of the system state in hand.

That characterization is not straightforward to produce. An E-CPSS continuously generates signals from all three of its spaces simultaneously: aviation safety databases, geospatial light pollution maps, and bird migration forecasts all exist for the ALAN context, but existing predictive modeling approaches treat these as an undifferentiated feature pool. The statistical outputs such approaches produce may be accurate, but they are not interpretable as descriptions of the tri-space system state. A designer reading a conventional feature importance ranking cannot tell whether the driving factors belong to the cyber, physical, or social space, and therefore cannot identify which space offers the most productive intervention surface. This interpretability gap is precisely what we address in this chapter.

We develop a CPSS-structured predictive modeling method in which we organize observational features according to the tri-space structure of the E-CPSS before any model is built. By partitioning features into cyber, physical, and social categories at the outset, and by interpreting model outputs through SHAP-based attribution mapped back to those same categories, we produce

state characterizations that are both predictively accurate and design-interpretable. The test problem is the impact of Artificial Light at Night on bird strike incidence in aviation, a co-evolutionary sociotechnical system where urban lighting infrastructure, FAA monitoring systems, and community governance behavior all interact to produce an emergent outcome that cannot be understood from any single space in isolation. Through this problem, we demonstrate that the CPSS-structured approach outperforms undifferentiated feature organization on interpretability while achieving comparable predictive accuracy, and we establish empirically that light pollution level is the strongest predictor of bird strike incidence, with implications that are actionable for urban planning policy and aviation safety governance.

The output of this chapter is I(si): the structured tri-space state characterization that forms the first formal component of the DSIM and feeds every subsequent layer. We address RQ1 in full. What we do not yet address is what to do with that characterization. When designers read the system state they learn its character, and they do not thereby learn what to do about it. In Chapter 4 we take up the decisions designers must make once they characterize a state, the coupling among those decisions, and the mathematics through which they quantify robustness under uncertainty.

Preface

Bird strikes on aircraft is a growing concern to wildlife preservation efforts and pose safety and economic challenges in aviation and urban design that amalgamate a sociotechnical system. Recent evidence suggests that artificial light at night (ALAN) is a contributing factor to bird strikes, especially during migration seasons, when birds are more likely to be disoriented by bright urban lighting. In this chapter, we explore the interplay of factors influencing bird strikes and present a machine learning-based predictive modeling approach informed by Cyber-Physical-Social Systems (CPSS). We emphasize the CPSS paradigm as a lens for modeling a range of systems engineering problems where evolving geospatial, temporal, and societal constraints characterize the environment. We underscore the interactions between the cyber, physical, and social spaces in model development and feature selection, providing a model-based system design with clear interpretations for public policy design and decision-making.

Through exploratory data analysis and predictive modeling, we identify significant patterns and trends in bird strikes and develop a model to predict bird strikes at a given time and location, with implications for urban planning, lighting design, and aviation safety. We emphasize the potential

of these predictive models to inform decisions in public policy and aerospace safety, aiming to mitigate bird strikes while accounting for ecological and industrial factors through model-based systems design. We envision this providing an actionable step toward designing Evolving Cyber-Physical-Social Systems (E-CPSS) and engineering informatics for public policy.

Research Question, Hypothesis and New Knowledge in this Chapter

Research Question	RQ1. What are the principles and the method for leveraging the cyber-physical-social structure of an E-CPSS in predictive modeling to uncover contributing factors and support design decisions?
Hypothesis	Hypothesis 1. Through a CPSS-structured feature organization that assigns observational variables to cyber, physical, and social spaces before model training, and SHAP-based importance attribution mapped back to those spaces after training, we construct interpretable tri-space state characterizations that reveal the contribution of ALAN to bird strikes and yield actionable insights for public policy and aviation safety decision-making.
New Knowledge	New Knowledge 1. A CPSS-structured predictive modeling method for constructing interpretable tri-space state characterizations of Evolving Cyber-Physical-Social Systems to support design decisions. Empirical confirmation that Artificial Light at Night is a statistically significant contributing factor to bird strikes, with light pollution level identified as the strongest predictor, with implications for public policy and aviation safety.

Keywords

Aviation Safety, Artificial Light at Night (ALAN), Cyber-Physical-Social System (CPSS), Predictive Modeling, Public Policy, Model-Based System Design, Sociotechnical System

3.1 Frame of Reference

Societal problems are characterized by a plethora of interconnected factors that collectively contribute to their complexity. To address these problems effectively, it is necessary to implement public policies anchored in informed approaches. In the intersection of ecological conservation, aviation safety, and urban design, the increasing threat of bird strikes and the adverse effects of Artificial Lights at Night (ALAN) necessitate a transdisciplinary approach guided by ‘systems engineering,’ ‘engineering informatics,’ and ‘design’ to guide public policies through decision options. Through model-based systems design, we demonstrate the interaction between the social and technical variables of the sociotechnical system, highlighting the utility of Cyber-Physical-Social Systems (CPSS) in addressing the complexities inherent in such interconnected issues. In this chapter, we extend our previous work presented at the ASME IDETC-CIE 2024 Conference [1].

Exemplifying a highly connected societal issue, bird strikes on airplanes, helicopters, and even spacecraft during takeoff are influenced by many anthropogenic and ecological variables and present a growing challenge in aviation safety and urban design. When examining the factors influencing bird strikes, a crucial aspect involves comprehending the interaction among ecological factors (driven by bird migration patterns and wildlife preservation), urban design, and aviation safety. Bird migration patterns and behavioral changes due to the impact of ALAN principally account for the social space, whereas wildlife preservation, urban design, and aviation safety incorporate social and physical spaces for model-based system design. Designers must therefore understand how these interactions work, so that they can decide how to reduce bird strikes and improve human safety. Since design choices may impact ALAN-induced bird strikes in future scenarios, we address this as a predictive model for systems design.

In this chapter, we aim to contribute to the progression of a machine learning-based predictive modeling approach in 'systems engineering', ‘engineering informatics’, and 'design' by addressing a challenge rooted in a complex system that involves 'birds as objects' to design public policies. We explore various domains of interaction with the overarching goal of designing enhanced policies. Before we present the method, we provide background on bird strikes." and "In this section we provide that background, identify the research gaps, and describe what designers can do with predictive modeling in policy design.

3.1.1 Overview of Bird Strikes and Their Significance

Bird strikes, defined as incidents where birds collide with a moving vehicle, have emerged as a concern for both wildlife conservation and aviation safety. Historically, these events were considered rare occurrences, but with the expansion of global air traffic and changes in natural habitats, they have become increasingly common, posing a threat to both avian populations and human safety.

Severe strikes can cause critical flight issues, posing safety risks to passengers and crew. In 2009, US Airlines Flight 1549 took off from New York and was shortly after struck by a flock of birds, resulting in the loss of all engine power. The pilot landed the flight successfully in the Hudson River with no injuries to passengers or crew, in an incident that quickly became known as “the Miracle on the Hudson" [2].

Furthermore, the significance of bird strikes extends beyond the immediate risk to flight safety. From an ecological perspective, these incidents highlight the broader issue of how human activities, such as aviation and urban development, can intersect with and impact wildlife. Birds, especially migratory species, face numerous threats from habitat loss, climate change, and urbanization. The specific challenge of *Artificial Light at Night* (ALAN) contributes to this by disorienting birds during their flight, making them more prone to collisions with aircraft [3].

The implications of bird strikes are multifaceted. Economically, they cause substantial damage to aircraft, leading to high repair costs and operational delays. Moreover, bird strikes have resulted in human injuries and fatalities, highlighting safety concerns in aviation. For wildlife, the loss of birds, particularly those from already declining species, exacerbates the challenges of conservation efforts [4]. Adaptation measures are expensive for both birds and people; implementing mitigation strategies demands significant financial and operational costs, while birds are otherwise forced to alter their behaviors and natural habitats.

This interaction of factors underscores the need for a multifaceted approach to addressing bird strikes. Understanding the role of ALAN in this context provides a tangible point of intervention. Thus, by examining the impact of artificial lights on bird behavior and the resulting increase in strike incidents, we can begin to design public policies and interventions that mitigate these risks.

The objective in designing such policies is twofold: first, to enhance aviation safety by reducing the frequency/severity of bird strikes, and second, to contribute to wildlife conservation efforts by minimizing one of the anthropogenic risks to bird populations.

3.1.2 Discussion of Artificial Light at Night (ALAN) as a Contributing Factor

Artificial Light at Night (ALAN) represents a pervasive environmental change affecting ecosystems worldwide. ALAN is the introduction of artificial light into previously dark nighttime environments; that light disrupts natural light cycles, profoundly influencing the behavior, physiology, and distribution of wildlife, including birds. The impact of ALAN on avian species is multifaceted, affecting their migration patterns, foraging behaviors, predator-prey interactions, and reproductive cycles as well [5].

The theoretical underpinnings linking ALAN to changes in bird behavior derive from the connection between light and the biological rhythms of many species. Birds, like many other organisms, rely on natural light cues for critical behaviors, including navigation and timing of their migrations. The disorientation caused by bright, artificial lighting, especially in urban areas, can change environmental cues that induce certain behaviors. This behavioral change can lead to increased incidences of bird strikes [5]. These strikes not only pose a significant threat to avian populations but also present safety and economic challenges in aviation and urban environments [4]. Researchers in this area have begun to shed light on the extent of ALAN's impact on wildlife. Studies have shown that nocturnally migrating birds, attracted or disoriented by intense light sources, are particularly at risk. This phenomenon is exacerbated during migration seasons when large numbers of birds are on the move, making them more vulnerable to collisions with illuminated structures [6].

3.1.3 Application to Public Policy Design and Mitigation Strategies

Through the approach we develop here, we help decision makers ground their policy choices in evidence. We consider three domains: ecological conservation, aviation safety, and urban development. This is carried out through careful consideration of multiple domains, including ecological conservation, aviation safety, and urban development. Our findings, derived from data exploration and random forest regression modelling, have potential applications in formulating bird-strike mitigation strategies. By identifying key predictors of bird strikes, aviation authorities

and wildlife conservationists can implement targeted measures, such as altered flight paths and lighting adjustments during high-risk periods identified by the model.

The utility of this model for public policy design carries implications for the formulation of bird-strike mitigation strategies. By pinpointing key predictors of bird strikes through our model and predicting them as the geographical location changes, aviation authorities and wildlife conservationists can deploy targeted engineering measures. For instance, the model's insights can inform flight path scheduling and facilitate adjustments to lighting systems during high-risk periods, thereby providing a tangible application of engineering solutions to mitigate the impact of bird strikes on aviation safety and ecological preservation.

Moreover, through the suggested model-based design, we offer designers an approach to grasp the interconnections among diverse variables within complex systems, facilitating a comprehensive understanding of interactions at the element level and their subsequent impact on the entire system. Through this model, we assist designers in comprehending the emergent effects stemming from the intricate interplay of multiple interactions. We can also automate this method computationally, so that designers build adaptive policy frameworks. These policies would dynamically evolve in response to fluctuations in variable parameters, guided by shifts in the environment that were not included in initial predictive variables.

This real-time adaptive approach is designed to support decision-making for aviation industries, urban planning departments, and wildlife conservationists. It enables them to make more informed decisions at the nexus of wildlife conservation and the prevention of inadvertent flight hazards. Through this research, we develop a suite of computationally informed solutions that policymakers use to mitigate the effects of Artificial Lights at Night through public policies.

3.1.4 Organization of the Chapter

Our focus in this chapter is on predictive modeling, aiming to bridge the gap between empirical evidence and actionable policies. By leveraging data on bird migration, light pollution, and bird strike records, we develop a model that can predict the likelihood of strikes at any given time and location. This predictive capacity is crucial for designing interventions that are both effective and efficient, targeting resources to times and places where the risk of bird strikes is highest. Through this work, we contribute to the ongoing dialogue between conservationists, aviation professionals,

and policymakers, seeking solutions that balance human needs with the imperative of wildlife preservation.

Organization of the Chapter

In **Section 3.2**, we provide a critical review of related work in geospatial and cyber-physical-social system (CPSS) modeling and illustrate the novelty of our approach.

In **Section 3.3**, we detail the methods employed in integrating multiple data sources and outline our approach to predictive modeling.

In **Section 3.4**, we present the results of our exploratory data analysis and the outcomes of our predictive modeling efforts, highlighting key findings and their implications.

In **Section 3.5**, we discuss the interpretation of our results, explore possible limitations of our method, and suggest directions for future research.

In **Section 3.6**, we offer closing remarks by summarizing the significance of our findings and their potential impact on public policy and conservation efforts.

In the following sections, we articulate and address the identified research gaps, contributing to a deeper understanding of ALAN's impact on bird strikes and offering actionable insights for mitigation strategies.

3.2 Review of Related Work

To model and predict bird strikes effectively, we propose examining the intersection of geospatial predictive modeling and the Cyber-Physical-Social Systems (CPSS) paradigm. We provide a critical review of existing literature by surveying current predictive modeling techniques for geospatial data and assessing their capabilities and limitations. Subsequently, we discuss the CPSS paradigm and explore how it has been applied to model emergent behavior arising from dynamic interactions within an environment characterized by cyber, physical, and social components. By synthesizing insights from these areas, we identify gaps in existing research and underscore the need for a novel modeling approach that can adapt to evolving geospatial conditions and societal demands.

3.2.1 Predictive Modeling for Geospatial Data

Predictive modeling for geospatial data has been extensively explored across various domains such as environmental monitoring, urban planning, and transportation systems, including in the context of wicked problems [7]. Traditional statistical methods like spatial regression and Kriging have been used to model spatial dependencies and predict outcomes based on geographical variables [8, 9]. With the advent of machine learning, techniques like decision trees, random forests, support vector machines, and neural networks have been applied to handle complex nonlinear relationships in spatial data [8].

For instance, random forests have been used to predict land-use changes [10], while neural networks have been employed to forecast weather patterns and urban growth [11]. In environmental science, spatiotemporal models have been developed to predict pollution dispersion and habitat suitability for wildlife [12].

Comparative Analysis of Existing Approaches

Researchers who use traditional statistical methods can interpret them and can model linear relationships with them, and they encounter difficulty with high-dimensional data and complex interactions [13]. Researchers obtain better predictive performance from machine learning models, and they often cannot see how those models reach a result [8]. With ensemble methods such as gradient boosting and random forests, researchers obtain higher accuracy and can still interpret part of the model [14].

Researchers have obtained good results on spatial data with convolutional neural networks, which represent spatial hierarchies, and they need large datasets and substantial computation to train them [15]. Researchers using existing models do not represent the changing interplay between social behavior [16] and the physical environment.

Limitations in Predicting Emergent Behavior

From our review of the literature, we posit that a significant limitation of current predictive models is their inability to capture emergent behavior arising from the complex interactions between the evolving physical environment and social behavior. Simon Elias Bibri notes this limitation of capturing the property of emergence in the context of smart cities, where a decision-support

framework for urban planning is proposed [7]. Similarly, Kovács-Győri and co-authors note the limitations of existing approaches for capturing the self-organizing aspect of urban systems, providing an example of stigmergy in which pedestrians build pathways that urban planners did not initially account for, subsequently inspiring the revision of structures based on new information patterns [8]. Kovács-Győri and co-authors, in this case, highlight the need for a geodata strategy, as alone these models typically are based on only static relationships captured over a given time frame, and thus often fail to adapt to changes over time or incorporate emergent social influences effectively. This limitation can hinder their applicability in scenarios where human behaviors, technological factors, and environmental conditions evolve and interact dynamically, such as in the case of bird strikes being influenced by urban lighting and subsequent mitigation measures.

3.2.2 CPSS as a Paradigm for Modeling Emergent Behavior

Cyber-Physical-Social Systems (CPSS) integrate computational resources (cyber), physical components (physical), and human behaviors (social) to create a holistic modeling framework. The CPSS paradigm has been applied in smart grids, intelligent transportation systems, and healthcare to model and manage complex systems effectively [17, 18].

In previous work, Allen and co-authors discuss the design of Evolving CPSS (E-CPSS), highlighting the need for models that can adapt over time and incorporate changes in technology, physical environment, and society [19]. This evolution is important for accurately predicting and managing emergent behaviors in complex systems. In predictive modeling, designers must therefore update the model over time as conditions change, and balance computational cost against practical constraints [16].

Approaches for Integrating Cyber, Physical, and Social Spaces

Integrating the cyber, physical, and social spaces of the CPSS requires methods that can handle heterogeneous data sources and dynamic interactions. Multi-agent systems have been used before, with agents representing entities within each space, and interacting according to predefined rules [20]. Machine learning models are also designed to process data streams and update predictions in real time.

Reza Alizadeh Kordabad addresses the management of computational complexity in CPSS [21] and illustrates an example of geospatial modeling relating to crime rate prediction in Los Angeles,

offering strategies to handle high dimensionality. We build on these approaches to model systems in which geospatial and societal factors change continually.

3.2.3 Research Gap Identification

Current geospatial predictive models are better in specific domains but, overall, often lack the inherent capacity to account for the dynamic and interdependent nature of evolving factors within their model representations [7, 8, 22]. Moreover, while CPSS offers a theoretical framework for such integration, practical applications in predictive modeling remain limited. Few studies have explicitly implemented a CPSS-based predictive technique, and fewer have applied them to real-world environmental and public policy concerns.

The Need for a Modeling Approach Applicable to Evolving Geospatial, Socially Informed Circumstances

Upon reviewing the literature, we suggest that a modeling approach is needed to handle the complexities of evolving geospatial data intertwined with social behaviors and technological influences. Such an approach could be universally applicable to problems characterized by dynamic interactions between societal factors and physical environments; we outline its potential generalizability in Section 3.5.3.

In the context of ALAN and bird strikes, our CPSS-based predictive model incorporates bird migration patterns (social/biological), urban lighting data (physical), and computational platforms (cyber) to predict and mitigate incidents most effectively. Moreover, if this model were continually updated with additional data over time, it would constitute an evolving CPSS. Through this integration, urban designers, aviation safety officials, and conservationists obtain attributions they can map to a named space and act on. We discuss our method in detail in Section 3.3 below.

3.3 Method: Impact of ALAN on Bird Strikes

In this section, we detail our method for model-based system design to develop policies aimed at mitigating the impact of ALAN on bird strikes, thereby achieving a parallel objective of safeguarding human lives from aircraft accidents. We describe our conceptual map, which illustrates the interconnectedness of the challenge within the CPSS paradigm.

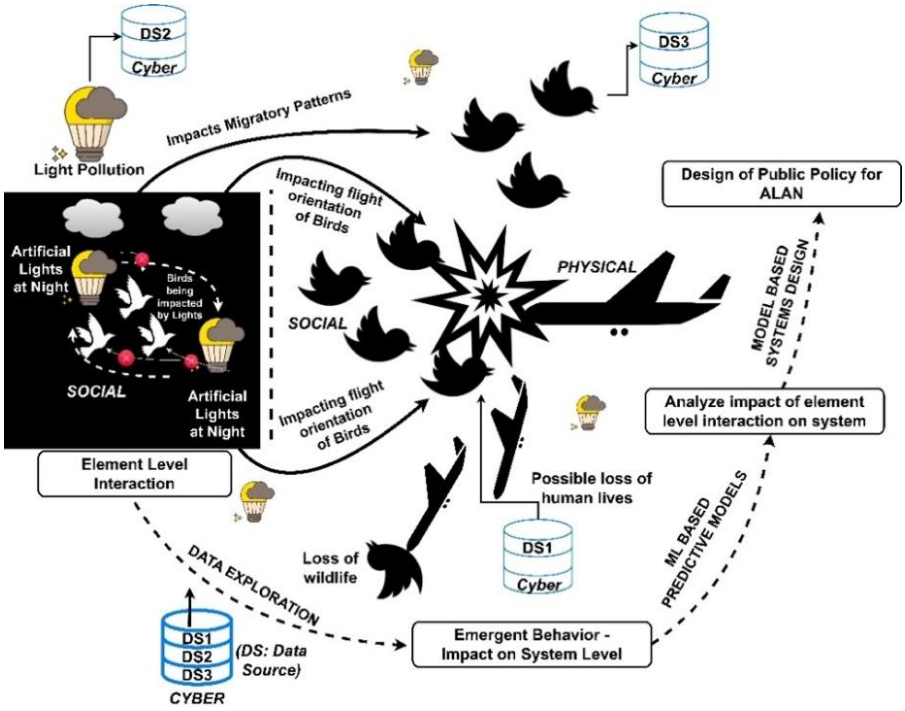


Figure 3.1: Conceptual Map: Policy design for the impact of ALAN on Birds as a CPSS

Impact of ALAN on Bird Strikes as a Complex System

In Figure 3.1, we illustrate our conceptual map for a model-based, data-driven system design approach to designing public policy aimed at mitigating the impact of Artificial Lights at Night (ALAN) on bird strikes and reducing potential aviation incidents. The influence of nighttime artificial lighting disrupts bird flights by disturbing melatonin production, leading to disorientation [5]. This light pollution not only affects flight orientation but may also induce variations in flight timing. Consequently, birds undergo behavioral changes, altering their migratory patterns and leading to detours, delayed departures, or stopovers in suitable habitats.

We approach this as a system with interacting component-level variables that cause system-level impacts through emergent behavior. Analyzing the interactive effects of light pollution on behavioral shifts in birds holistically is imperative. The focus extends beyond the critical analysis of ALAN's direct impact on bird strikes; it also encompasses an understanding of emergent effects arising from interactions among various variables. Further elucidation of this approach is provided in the subsequent sections, delving into the intricacies of systems engineering principles and policy design considerations. Through the proposed approach, we aim to do the following:

1. Analyze the interaction between component-level variables and assess their impact on the emergent behavior of the system.
2. Analyze interactions between transdisciplinary variables comprising diverse social and technical domains.
3. Utilize model-based system design for better-informed decision-making.
4. Investigate and discern the primary variables with the most significant impact on bird strikes.

In this chapter we integrate multiple data sources to investigate the impact of Artificial Light at Night (ALAN) on bird strikes and train a model to predict strikes, leveraging the FAA Wildlife Strike Database (DS1 in Figure 3.1), a light pollution GIS map (DS2 in Figure 3.1), and BirdCast migration forecasts (DS3 in Figure 3.1).

Addressing ALAN as model-based system design

We introduce our model-based system design approach in Figure 3.2, where a spectrum of variables provides invaluable insights for system design. In this intricate system, the exchange of information is not immediately apparent, given the diverse social and technical domains encompassing aviation, wildlife, and light emission. However, our method accounts for the nuanced interactions among these variables through robust data exploration methods. The primary aim is to empower policymakers by providing comprehensive, data-driven insights derived from our model-based system design. These insights, derived from the intricate web of interconnected variables, serve as a cornerstone for designing well-informed policies that harmonize the complexities of aviation, wildlife preservation, and urban lighting.

This model-based system design, as shown in Figure 3.2, is supported by the conceptual map that is presented in Figure 3.1. This is embodied in the CPSS paradigm. The aircraft, represented in Figure 3.1, constitute the major elements of physical space, whereas light pollution, birds, and changes in their behavior are the primary entities of social space. The geographical and urban planning factors contribute to both social and physical spaces. The data sources presented as DS1, DS2, and DS3 in Figure 3.1 and explained in detail in Section 3.3.1 form the cyber space that enables data-driven design for this model-based system.

In Figure 3.3, we outline our approach to predictive modeling through a sequence of steps including problem identification, data source selection, exploratory data analysis (EDA), and predictive model construction. This is further expanded in the following sections.

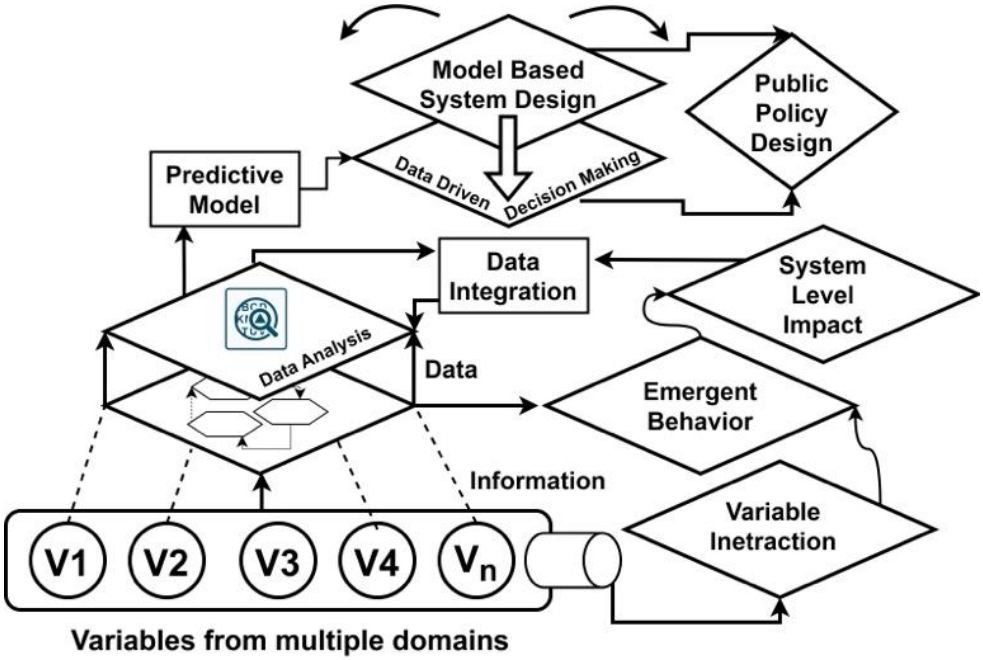


Figure 3.2: Addressing Bird Strikes through a Model-Based, Data-Driven System Design

3.3.1 Description of Data Sources

We use the FAA Wildlife Strike Database to provide initial exploratory data analysis (EDA) findings on the distribution of bird strikes, including trend analysis, species-specific analysis, and temporal patterns (Section 3.4.1). In Sections 3.4.2 and 3.4.3, we incorporate the light pollution GIS map and BirdCast migration forecasts to perform statistical tests and build a model to predict bird strike occurrences. By examining the confluence of these data sources, we seek to contribute to a broader understanding of ALAN's impacts, with a specific focus on the role of spatiotemporal predictive modelling in informing public policy design.

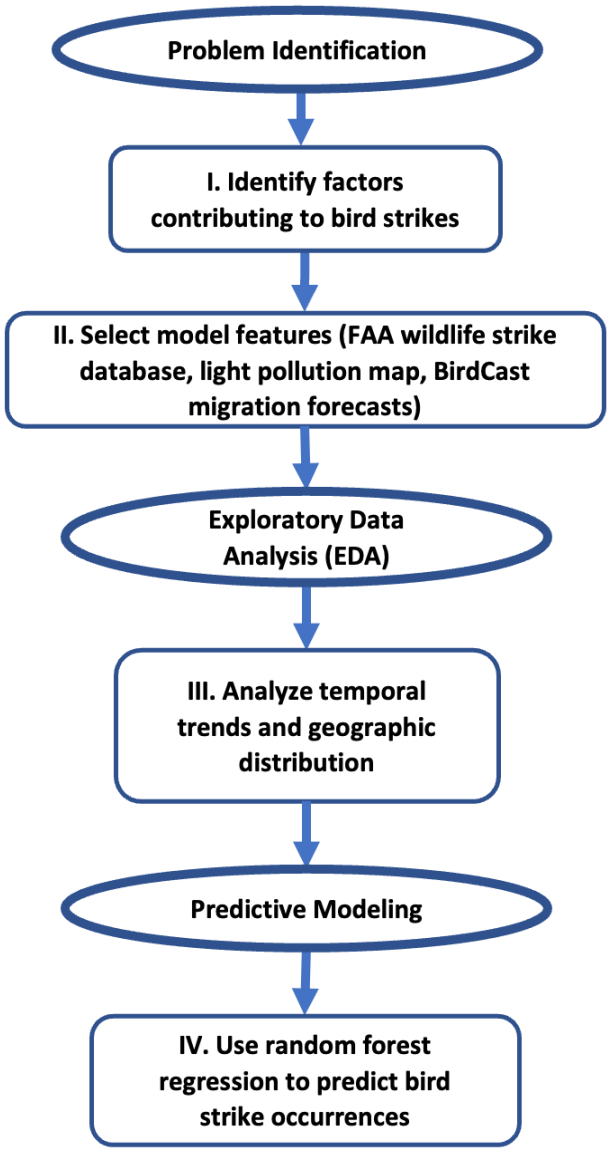


Figure 3.3: Diagram of Predictive Modeling Approach

FAA Wildlife Strike Database

The Federal Aviation Administration Wildlife Strike Database [23] provides a comprehensive record of bird strike incidents reported to the FAA, with over 286,000 reported strikes from 1990 through 2023. Approximately 97% of all recorded strikes with aircraft in the United States involve birds, though strikes with other animals such as deer and coyotes have also been reported [24]. Bird strikes are voluntarily reported, primarily by pilots and airport personnel. Records generally

include details on the date, time, location, species involved, and the extent of damage to the aircraft. This dataset serves as the foundation of our analysis, offering insight into the frequency and distribution of bird strikes in the United States.

GIS Light Pollution Map

Light pollution data is represented through a Geographic Information System (GIS) raster that maps the intensity and distribution of light pollution across North America [25, 26], sourced from NASA Visible Infrared Imaging Radiometer Suite (VIIRS) Day/Night Band VN2 yearly compositive of cloud-free satellite data in 2020. This map includes color bands ranging from 0 to 14, with each band corresponding to a discrete light pollution category. In Table 3.1, we provide unit conversions of raster bands to both magnitudes per square arcsecond and brightness ratio, based on the color scale derived from David Lorenz's Light Pollution Atlas [25]. The magnitudes per square arcsecond reflect standard astronomical brightness, while the artificial/natural brightness ratio provides a comparative measure of artificial light pollution. We use this data to quantify light pollution levels at each bird-strike location, enabling us to perform correlation analyses and train a predictive model.

Table 3.1: Dual-unit Representation of Light Pollution Data [25]

Raster Band	Magnitudes per Square Arcsecond (mag/arcsec^2)	Brightness Ratio (Artificial/Natural)
0	22.00 – 21.99	< 0.01
1	21.99 – 21.93	0.01 – 0.06
2	21.93 – 21.89	0.06 – 0.11
3	21.89 – 21.81	0.11 – 0.19
4	21.81 – 21.69	0.19 – 0.33
5	21.69 – 21.51	0.33 – 0.58
6	21.51 – 21.25	0.58 – 1.00
7	21.25 – 20.91	1.00 – 1.73
8	20.91 – 20.49	1.73 – 3.00
9	20.49 – 20.02	3.00 – 5.20
10	20.02 – 19.50	5.20 – 9.00

11	19.50 – 18.95	9.00 – 15.59
12	18.95 – 18.38	15.59 – 27.00
13	18.38 – 17.80	27.00 – 46.77
14	< 17.80	> 46.77

BirdCast Migration Forecasts

The BirdCast project entails forecasts of bird migration in North America, providing valuable data on the movement of birds, especially during peak migration periods, using input data from the NEXRAD weather surveillance radar network across the continental United States [27, 28]. Data for this project were graciously provided by Alexander Tedeschi, Andrew Farnsworth, Adriaan Dokter, and Julia Wang at the Cornell Lab of Ornithology. We use 2023 forecast data, captured each day at 0600Z, to train our predictive model. These data are divided into GeoTIFF files for each month, with each layer containing a time dimension. For any single day, there are 4 raster bands:

- *U (birds'* ground speed component west to east in m/s)
- *V (birds'* ground speed component south to north in m/s)
- *MTR* (migration traffic rate in birds/km)
- *VID* (vertically integrated density in birds/km^3)

Moreover, there are 4 extra bands corresponding to the *Kriging* variance of each variable. Kriging variance in the context of BirdCast reflects the uncertainty in predictions of bird migration intensity across different geographic locations. BirdCast uses kriging, a geostatistical interpolation technique, to estimate bird migration patterns from sparse observational data; the NEXRAD network does not have complete coverage of all airspace. The Kriging variance provides a measure of the expected error or uncertainty of these estimates, indicating how much confidence we can have in the predicted migration intensities at locations where direct observations are not available [29]. This helps researchers to better understand and predict bird movements

3.3.2 Methods for Preparing and Integrating Data

The foundational step in our Exploratory Data Analysis (EDA) and modeling involved preparing and integrating three data sources: the FAA Wildlife Strike Database, a light pollution GIS map, and BirdCast migration forecasts. We follow a systematic approach when we combine the data, assess data integrity, and prepare the dataset for analysis and predictive modeling.

Geo-referencing Light Pollution GIS Raster

To integrate the light pollution data with the bird strike database effectively, the light pollution GIS raster was georeferenced [30] using QGIS software. Georeferencing involved assigning real-world coordinates to each pixel of the raster image, aligning with the WGS 84 - EPSG:4326 coordinate reference system. This step enables the spatial joining of light pollution data with bird strike incidents by matching latitude and longitude values, providing geospatial context for the analysis.

BirdCast Migration Forecast Integration

The BirdCast migration forecasts, offering insights into bird migration patterns, were integrated by leveraging Python's GIS libraries (rasterio and pyproj) to process GeoTIFF files. This integration allowed us to join datasets by latitude, longitude, and day of the year, enriching the analysis with temporal and spatial dimensions of bird migration in relation to light pollution.

3.3.3 Exploratory Data Analysis (EDA) Techniques

Our EDA begins with a deep dive into the FAA wildlife strike dataset, aiming to unveil patterns and anomalies that could inform our subsequent predictive modeling efforts. Using Python libraries such as pandas and matplotlib, we conduct a series of analyses focusing on temporal trends, geographical distributions, and species-specific strike rates.

- **Temporal Trends**: We analyze bird strike occurrences over time, identifying peak periods that correspond with migratory seasons [31]. Through this analysis, we observe a notable increase in strikes during spring and fall migrations.
- **Geographic Distribution**: Utilizing GIS tools, we map bird strikes against light pollution levels, uncovering hotspots where high ALAN levels coincide with increased strike incidences. This spatial analysis highlighted urban areas as zones of elevated risk.

- **Species-Specific Analysis**: By examining the species involved in reported strikes, we discern patterns indicating that certain migratory birds are more susceptible to ALAN-related incidents. This insight underscores the need for species-specific strike-mitigation strategies.

In Sections 3.4.1 and 3.4.2, we use maps, bar plots, and line plots to represent spatial and temporal relationships between bird strikes and light pollution. These visual tools enhance our understanding of the data and guide our model design choices, detailed in Section 3.3.4.

3.3.4 Statistical and Machine Learning Techniques for Predictive Modeling

Building on the foundational data joins and EDA detailed in the preceding sections, we employ a combination of statistical tests and machine learning techniques to understand the impact of ALAN and to develop a predictive model of bird-strike incidents. Through this approach, we aim to harness the predictive power of our integrated datasets to forecast bird-strike risk with greater precision and identify effective mitigation strategies.

Correlation Analysis

Kendall's τ and Spearman's ρ rank correlation coefficients are calculated to explore the strength and direction of the relationship between bird strikes and light pollution levels [32]. This allows us to quantify the impact of ALAN on bird strikes and determine its predictive strength in model construction.

Predictive Modeling with Random Forest Regression

Focusing on a random forest regression model, implemented using Python's scikit-learn library [33], allows us to model the continuous outcome variable representing the frequency of bird strikes under varying conditions. We selected this model because we sought to balance predictive performance with interpretability and because the modelers configured it to represent non-linear relationships between the predictors and the target variable [34].

- **Model Development**: The random forest was trained using features identified as significant through our statistical tests, including time of day and light pollution level.
- **Feature Importance**: Using our model, we provide insight into feature importance, highlighting which variables most significantly affect bird strike rates and aiding in the prioritization of mitigation efforts.

Table 3.2. Training Dataset for Predictive Model

Number of Strikes (y)	Month of the Year (X)	Day of the Month (X)	Latitude (X)
Longitude (X)	Light Pollution Level (X)	(MTR) Migration Traffic Rate (X)	(VID) Bird Density (X)

The training dataset, as outlined in Table 3.2, contains one dependent variable (y) and seven predictors (X). Data are grouped by latitude, longitude, and day of the year, for a total of 72,545 records in the dataset. Moreover, in Table 3.3, we outline the ranges and distributions of key variables, illustrating their connections. The mean number of strikes at a given location, for instance, is 3.29, with a range between 1 and 99. Additionally, the range of latitude and longitude values demarcates the geographic scope of our dataset.

Table 3.3: List of variables and their distribution

Count: 72,545	Mean	Standard Deviation	Minimum Value	Maximum Value
Number of Strikes (y)	3.29	4.69	1	99
Latitude (X)	37.4503	5.54	24.5561	49.1939
Light Pollution Level (X)	11.69	1.77	0	14
Migration Traffic Rate (X)	881.56	2,830.06	0	79147.52
Bird Density (X)	22.69	58.42	0	1368.47

This listing helps illuminate how each variable contributes to our model, the inter-variable relationships, and the model's collective influence on accuracy. Understanding the variability within these variables is useful for evaluating the fidelity of our predictive model and the interpretability of performance metrics, as it affects both the training process and the generalizability of our predictions.

Evaluation and Validation

To ensure the reliability and applicability of our model, we implement the following evaluation and validation strategies:

- **Performance Metrics:** Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R-squared (R^2), and Weighted Mean Absolute Percentage Error (WMAPE) are used to quantify the model's accuracy in predicting bird strike rates, offering clear measures of predictive performance.
- **Cross-Validation**: We implemented k-fold cross-validation with k=5 to assess the model's generalizability across different subsets of the data, ensuring robustness against overfitting.
- **Hyperparameter Tuning**: To avoid overfitting and enhance predictive accuracy, we employed 250 decision trees in the random forest and tuned the maximum tree depth to 25, the minimum samples for splitting to 2, and the minimum samples per leaf to 5. The model was developed using an 80-20 train-test split.

3.4 Results

Our focus in exploratory data analysis (EDA) is on temporal trends and the geographic distribution of bird strike occurrences, and through our predictive model policymakers can estimate strike counts from environmental, temporal, and geographic predictors.

3.4.1 Exploratory Data Analysis (EDA) Findings

Our EDA focused on extracting patterns and trends from the FAA wildlife strike dataset. Below, we discuss the findings revealed in the figures.

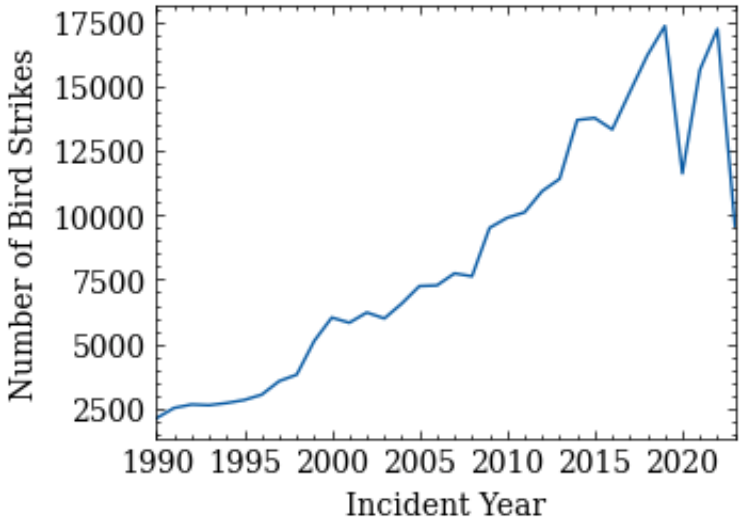


Figure 3.4: Yearly Bird Strikes Trend (1990-2023)

In Figure 3.4, we observe an overall upward trend in bird strikes from 1990 to 2023, with the exception of 2020, a year with significantly reduced airport traffic due to COVID-19-related public transportation protocols. s strikes have increased, aviation safety officials and wildlife managers face a growing challenge, and we recommend that researchers investigate contributing factors such as environmental change and growth in air traffic.

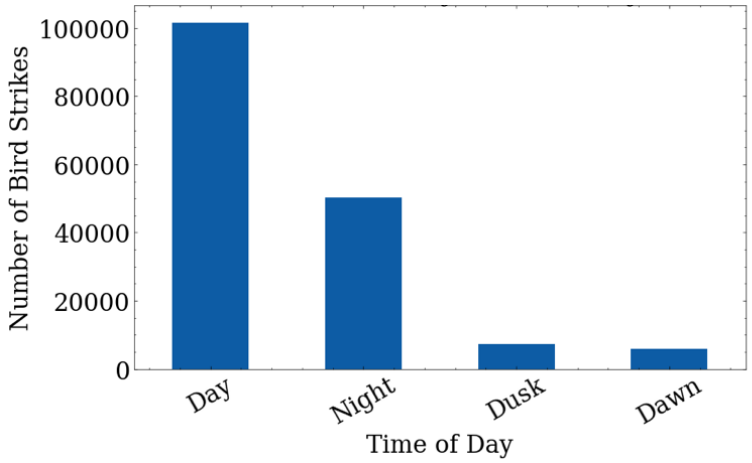


Figure 3.5: Recorded Bird Strikes by Time of Day (1990-2023)

In Figure 3.5, we observe differences in voluntarily reported time-of-day data for bird strikes in the FAA database, with four discrete categories including “Day”, “Night”, “Dusk”, and “Dawn”. We observe that 61.5% of bird strikes occurred during the daytime, 30.4% at night, and 8.1% at dawn or dusk.

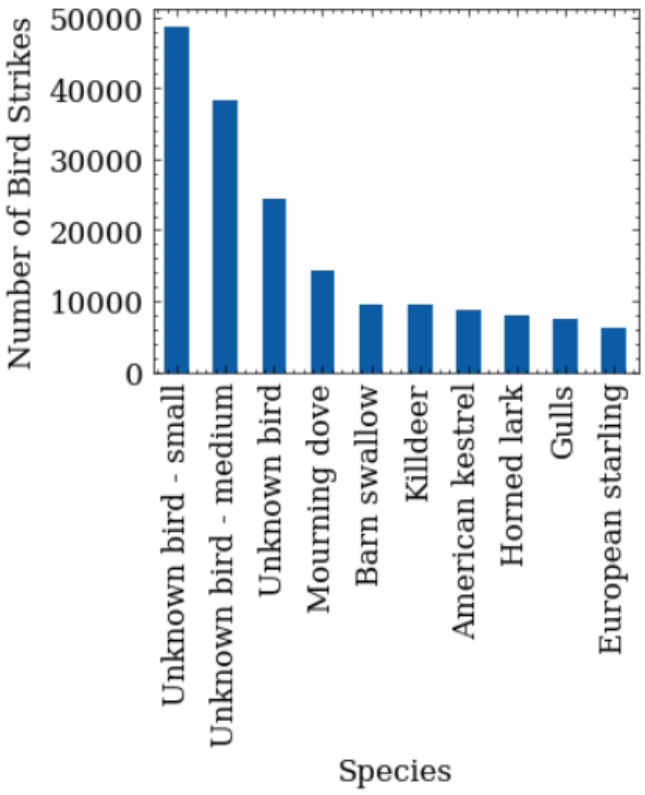


Figure 3.6: Species-Specific Analysis (Top 10 Species Involved in Bird Strikes)

Through our species-specific analysis in Figure 3.6, we observe that certain bird species are more frequently involved in strikes. The most frequently represented values in the bar plot are “Unknown bird – small”, “Unknown bird – medium”, and “Unknown bird”. Following these are specific species, including mourning dove, barn swallow, kildeer, and American kestrel. We observe that smaller bird species are most represented in this bar plot and are more commonly affected by aviation-related bird strikes.

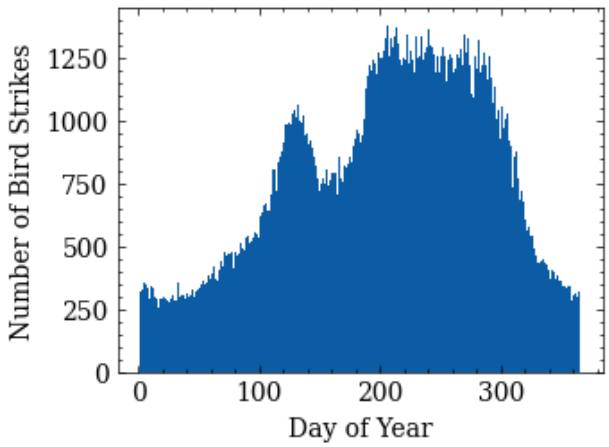


Figure 3.7: Distribution of Bird Strike Occurrences by Day of Year (1990-2023)

In Figure 3.7, time-series analysis across the day of the year from 1990 to 2023 illustrates seasonal trends in bird strikes, with peaks during migration periods in spring and fall. We attribute this pattern to the migratory behavior of bird populations, and we recommend that airport operators and wildlife managers plan for the migration seasons.

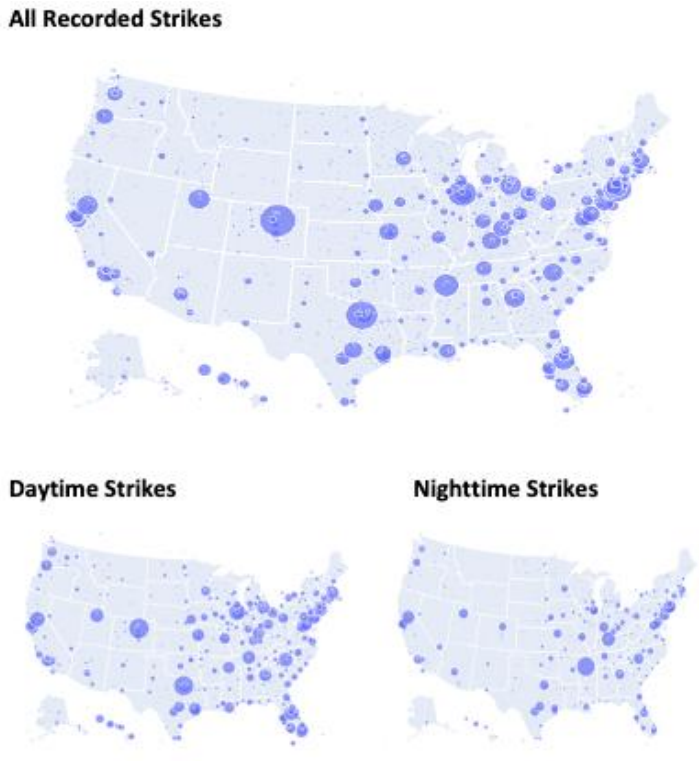


Figure 3.8: Recorded bird strike locations in the United States (1990-2023)

In Figure 3.8, we present a geographic analysis of bird strikes, revealing hotspots across the United States. Larger circles indicate a higher number of bird strikes at a given location. We identify airports near migration routes and urban centers as the most exposed. We also observe noteworthy differences between daytime and nighttime strikes. This spatial analysis is critical for prioritizing intervention areas and guiding the deployment of resources for bird-strike prevention.

3.4.2 Correlation Analysis of Bird Strikes and Light Pollution

In this section we present the findings from our correlation analysis, focusing on Kendall's τ and Spearman's ρ rank correlation coefficients to investigate the relationship between light pollution and bird strike incidents.

- Kendall's τ coefficient is calculated to be 0.33, with a p-value less than 0.01, indicating a statistically significant correlation that is not due to random chance.
- Spearman's ρ correlation coefficient is calculated to be 0.42, with a p-value less than 0.001, indicating a statistically significant correlation. This outcome corroborates the findings from Kendall's τ, highlighting a robust association between variables.

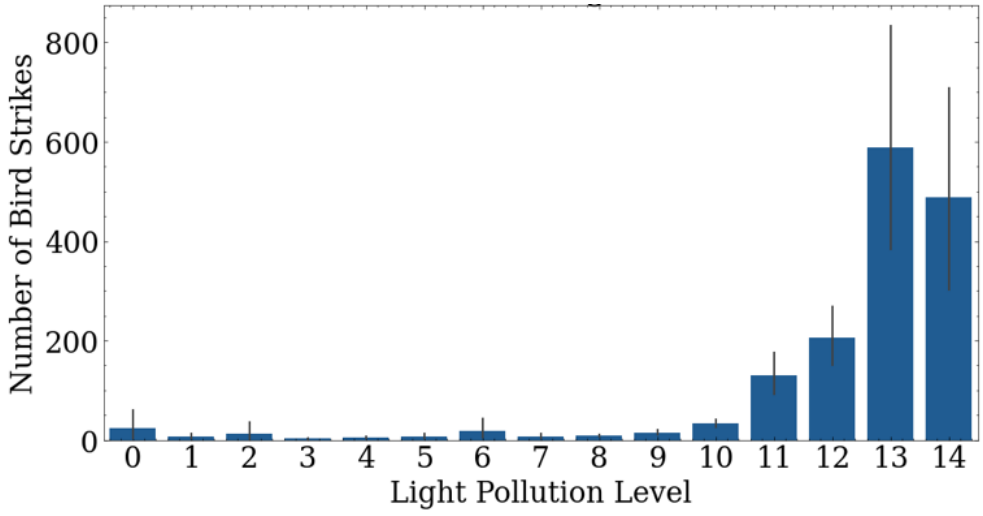


Figure 3.9: Bar Plot of Light Pollution Levels Against Bird Strike Count

In Figure 3.9, we plot discrete light pollution level categories (0-14) against strike records in the FAA wildlife strike database, observing higher strike counts as light pollution levels increase (see Table 3.1 for conversions to mag/arcsec² and the artificial light brightness ratio). These findings are in alignment of our correlation tests and suggest that light pollution is a significant factor influencing bird strikes, with higher levels of light pollution associated with higher rates of these incidents.

3.4.3 Predictive Modeling Outcomes

In this sub-section, we discuss the outcomes of the random forest regression model and its potential applicability in a public policy design context.

Model Performance Metrics

- **Mean Squared Error** (MSE): 4.16
- **Root Mean Squared Error** (RMSE): 2.04
- **Mean Absolute Error** (MAE): 1.16
- **R-squared** (R^2): 0.80
- **Weighted Mean Absolute Percentage Error** (WMAPE): 35.40%

From these metrics we read the model's average error and its fit. With $R^2 = 0.80$ on the held-out test set, the model accounts for about 80% of the variance in recorded strike counts.

K-fold Cross-validation

Using K-folds cross-validation, we evaluate the model across different subsets of the dataset to minimize the potential bias of a single train-test split. The average MSE and MAE across the folds were 4.02 and 1.15, respectively. These fold-level averages fall close to our single-split evaluation, so the model's error does not vary much across subsets of the data.

Monthly Performance Metrics

In Table 3.4, we provide monthly averages of the model's performance metrics across all years in the training data. The average absolute and squared errors vary from month to month, indicating

differences in prediction accuracy over time. For example, January has an average absolute error of 0.62 and a squared error of 1.15, while May has slightly higher errors, with an average absolute error of 0.99 and a squared error of 3.32.

Table 3.4: Monthly averages of performance metrics in the training dataset

Month	**Average Absolute Error**	**Average Squared Error**
January	0.62	1.15
February	0.60	0.99
March	0.71	1.31
April	0.89	2.41
May	0.99	3.32
June	0.88	2.50
July	1.03	3.23
August	1.05	3.08
September	1.05	3.41
October	1.03	3.45
November	0.87	2.32
December	0.68	1.33

Overall, across the monthly absolute errors and absolute squared errors in the dataset:

- The **average absolute error** is approximately 0.87, suggesting that, on average, model predictions deviate from the actual number of strikes by about 0.87 strikes.
- The **average squared error** is approximately 2.38, reflecting the average squared deviation of the predictions from the actual values, adjusted for the scale of the dataset.

Relevance to Public Policy Design

Through Figures 10, 11, and 12, we interpret the random forest model via evaluations of permutation importance, feature importance, and SHAP values (SHapley Additive exPlanations).

Based on the contributions of each feature to the model, we observe that light pollution has the strongest predictive value across two of the three evaluation criteria (permutation importance in Figure 3.10 and SHAP values in Figure 3.12). In the SHAP beeswarm plot we see that observations with high light pollution values (in red) carry positive SHAP values, so the model predicts more strikes at those locations.

Additionally, latitude, longitude, and month are the strongest predictors by measure of feature importance in Figure 3.11. We also observe that day, bird density, and migration traffic rate have comparatively low impact across all selected evaluation criteria.

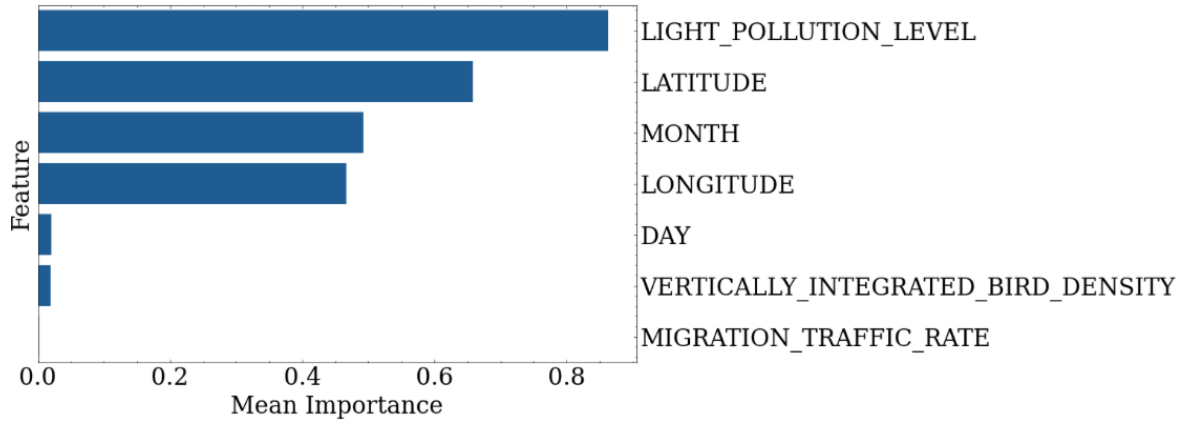


Figure 3.10: Permutation Importances in Random Forest Model

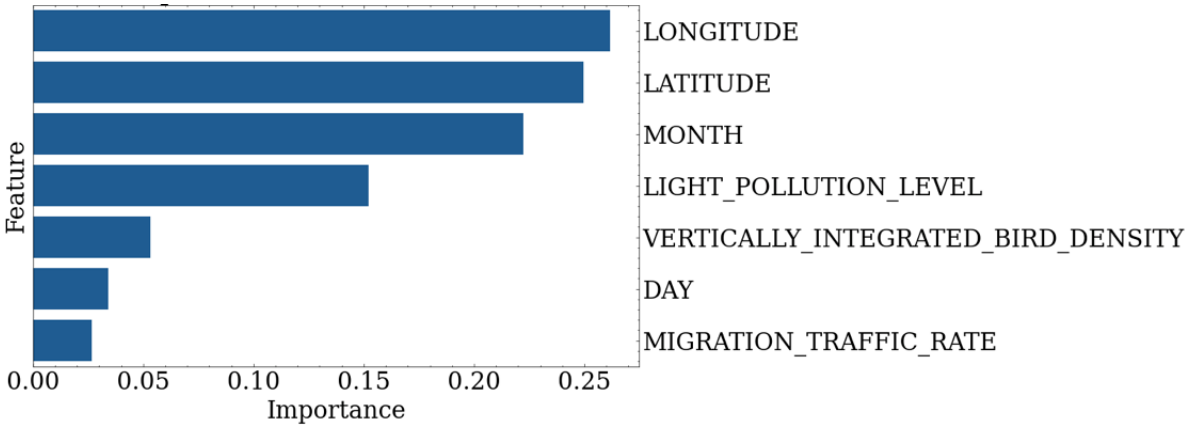


Figure 3.11: Feature Importances in Random Forest Model

From the model's predictions, policymakers and conservationists can act. Using the model, we identify the time frames and geographic areas in which we estimate elevated strike risk, so that airport operators and city officials can act before the season begins. Policymakers can then design policies that address urban development, aviation safety, and wildlife conservation together.

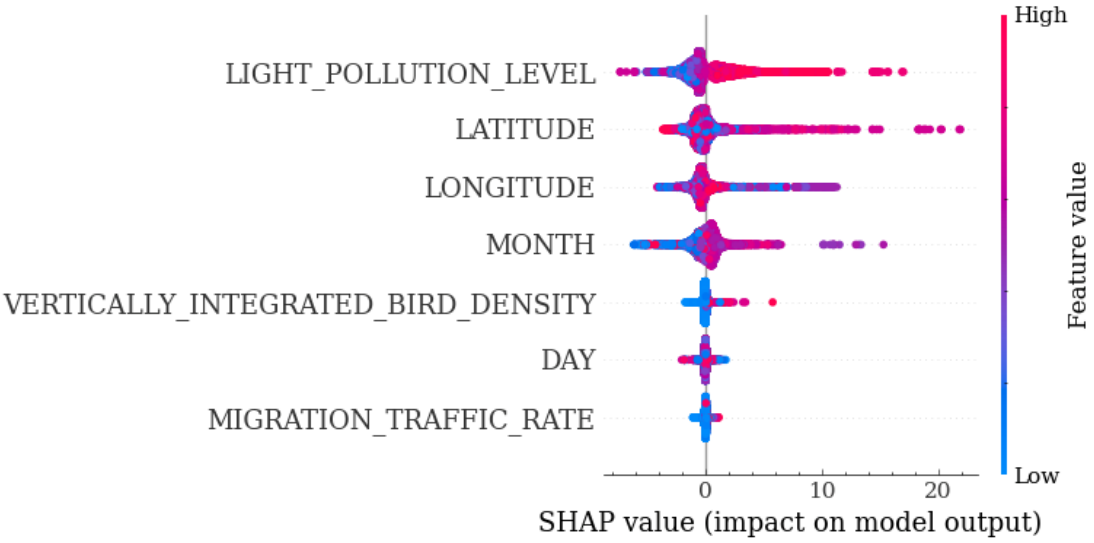


Figure 3.12: SHAP Values in Random Forest Model (Beeswarm Plot)

3.5 Discussion

In this section, we interpret the results of our data analysis and predictive modeling efforts, discuss their applicability to public policy design, and outline directions for future work.

3.5.1 Results Interpretation

From our investigation and our predictive model, we draw conclusions about wildlife preservation in landscapes that people have modified. By elucidating the relationship between ALAN and bird strikes, we aim to shed light on broader ecological and safety implications, offering a foundation for informed public policy and urban planning decisions.

Contextualizing within Aviation Safety and Urban Planning

Through our analysis, we identify a correlation between increased light pollution and the incidence of bird strikes. We interpret this correlation as evidence of risks to avian populations and of safety and economic challenges for people in the aviation industry. By mapping bird-strike occurrences

against light pollution levels, we have identified high-risk zones that often coincide with urban areas and major flight paths. Through this, we suggest that mitigating the effects of ALAN could significantly enhance aviation safety by reducing the frequency of bird strikes.

Moreover, based on the temporal and geographic patterns identified in our EDA, we emphasize the need to consider bird migration dynamics in urban planning and aviation safety strategies. Because we record peaks during the spring and fall migration seasons, we recommend that city officials dim or redirect lighting during those weeks.

Practical Applications for Designing Public Policy

The model-based system design that leverages predictive modeling is a potential tool for policymakers, enabling forecasts of bird strike occurrences at any given time and location with a high degree of accuracy. Through this approach, designers will be able to advance principles of systems engineering and model-based system design to develop effective public policies. By integrating various data sources and framing them within the CPSS paradigm, we provide a basis for selecting relevant predictors of bird strikes. This anchors in the idea of information models for social problems, where the focus is on the effective use of multiple information sources and their integration to enable designers to understand the causes of emergent system behavior, resulting in fundamental developments in the design of systems for social problems. Through this approach designers can target mitigation strategies:

- Incorporating predictive modeling into urban planning processes to implement mitigation efforts at high-risk locations and timeframes.
- Adjusting urban lighting schemes during migration periods to reduce attraction or disorientation of migratory birds.
- Planning new developments or modifying existing structures to mitigate the impact of light pollution on surrounding ecosystems.

Our findings are relevant to developing public policies that balance human needs with wildlife conservation, as demonstrated through these practical applications. By informing such policies with predictive modeling, we can proactively address the challenges posed by ALAN, enhancing both aviation safety and biodiversity conservation.

3.5.2 Limitations

In this chapter, we contribute new insights into the relationship between Artificial Light at Night (ALAN) and bird strikes, utilizing predictive modeling and cyber-physical-social system design to inform public policy and conservation strategies. We state four limitations of our study, each of which bears on how far a designer can carry our results.

Analysis of General Light Pollution Rather Than Specific Light Sources

One limitation of our analysis is that it does not discriminate between different types of lighting sources. Specifically, our GIS light pollution map captures light pollution observed from satellite data, including artificial lighting sources and industrial sources such as flaring from oil and gas sites [35]. This means that our correlation analysis and predictive model do not directly incorporate ALAN as a variable, but instead use general satellite-observed light pollution to approximate its impact. In future work, researchers can refine the model by controlling for industrial light sources such as gas flaring, and thereby assess the effect of ALAN more directly.

Feature Selection in Predictive Model

Our predictive model, while robust, is constrained by the variables included. Factors such as altitude and airport traffic can also significantly influence the occurrence of bird strikes. Including these additional predictors may enhance the model's accuracy and applicability. Researchers may consider integrating these and other relevant variables, especially in untested environments outside the geographic scope of our research, to construct a comprehensive model that captures additional factors influencing bird strikes.

Assessing the Performance of the Model Over Time

As the incidence of bird strikes may decrease due to effective interventions informed by our model, the resulting data sparsity could challenge the model’s ongoing accuracy and relevance. The diminishing frequency of bird strikes, while a desired outcome, may lead to the underrepresentation of certain conditions or outliers in future datasets, potentially skewing the model predictions. To counteract these challenges, strategies such as integrating synthetic data could be employed to enhance the dataset, ensuring robust predictive capabilities despite fewer real-world incidents. Additionally, implementing adaptive feedback mechanisms that recalibrate

the model based on the outcomes of mitigation efforts can help maintain accuracy, as noted in the design method for Evolving CPSS (Section 3.2.2). Through these approaches, we can ensure that our model remains dynamic, responsive to change, and facilitates continuous improvement and relevance in predicting bird strikes under varied conditions.

Other Considerations and Areas for Future Work

- Spatial and Temporal Resolution: The granularity of data used in our analysis might not capture variations in light pollution (2020 data) or bird migration patterns (1/day forecast data) over the time period of bird strikes we studied (1990-2023). Higher-resolution data could provide more precise insights into the spatial and temporal dynamics of bird strikes.
- Voluntary Reporting Bias: Reliance on the FAA Wildlife Strike Database, which is based on voluntary reporting, may introduce bias. Not all bird strikes are reported, especially those that do not result in significant damage. Efforts to adopt more systematic data-collection methods could reduce this bias.
- Generalizability Across Regions: Our study focuses on the United States, and while the model shows promise, its applicability to other geographical regions remains to be tested. Different migratory patterns, species diversity, and urban lighting standards could affect the model's relevance in other contexts.

3.5.3 Generalizability of the Modeling Approach

In exploring the interplay between artificial light at night (ALAN) and bird strikes, we demonstrate through our modeling approach the effectiveness in predicting complex phenomena driven by dynamic interactions among model representations. Through the salient features of this method, grounded in the Cyber-Physical-Social Systems (CPSS) paradigm and a model-based system design for geospatial data, we present opportunities for application across various domains facing similar challenges. In this section, we highlight these features and discuss their applicability in example scenarios relating to adaptive manufacturing systems, ecology, and civil engineering.

Salient Features of the Method

We posit that our modeling approach is characterized by several key features that contribute to its adaptability and effectiveness in complex systems:

- **Holistic Integration** We provide relevant model features by combining cyber, physical, and social components to capture complex interactions and emergent behaviors.
- **Model-Based System Design** The focus is on understanding the system's dynamics and relationships to inform model construction [36] and policy design.
- **Geospatial and Temporal Modeling** incorporates spatial and temporal data to enhance predictive accuracy and relevance [37].
- **Potential for Continued Adaptability with E-CPSS** allows the model to evolve over time with new data, maintaining accuracy in changing environments.
- **Actionable Insights for Decision-Making** provide interpretable results that inform policy and design interventions across domains.

Extended Modeling Application: Adaptive Manufacturing Systems

Adaptive manufacturing systems [38, 39] operate in environments characterized by fluctuating demand, supply chain uncertainties, and evolving consumer behavior, which create emergent complexities akin to those in our bird strike setting. Thus, applying our modeling approach in this domain has the potential to enhance responsiveness and efficiency.

In the context of adaptive manufacturing, the CPSS paradigm may be loosely mapped as follows, depending on enterprise needs: 1. The cyber component includes data analytics platforms that process real-time market data, inventory levels, and production metrics; 2. The physical component encompasses manufacturing equipment, production processes, and logistics networks; and 3. The social component involves market trends, regulatory environments, and stakeholder objectives.

Integration with geospatial components involves incorporating location-specific data, such as regional demand variations, distribution routes, and supplier locations. Temporal dynamics are also accounted for by factoring in seasonal demand and market trends. For example, manufacturers of seasonal goods can integrate historical sales data and economic indicators to accurately forecast demand, enabling them to adjust manufacturing schedules and inventory levels in real time, thereby reducing the costs of overstocking or understocking. Similar approaches for just-in-time manufacturing, including digital twins and real-time analytics, have been discussed in the literature

[36, 40, 41]. Although our method also offers the added benefit of providing a framework for modeling emergent behavior with geospatial data, as identified in the research gaps.

The Evolving CPSS concept plays a crucial role in this adaptive manufacturing scenario by allowing models to update with new sales data, market indicators, and supply chain information. This dynamic adaptation helps anticipate changes in demand driven by factors such as inflation, production challenges, or shifting consumer preferences. As a result, manufacturers can respond to unexpected events by dynamically updating models and strategies.

Applying our approach in adaptive manufacturing could offer several benefits, including increased efficiency by aligning production closely with actual demand, enhanced responsiveness by quickly adapting to market changes, and improved decision-making through data-driven insights to inform strategic planning.

Extended Modeling Application: Engineering

Engineers and ecologists who work where wildlife movement meets human infrastructure face complex challenges, and they could use our approach. Predictive models have previously been used in informing the design and placement of structures like wildlife crossings [42], and our method may help support the long-term efficacy and adaptability of these approaches.

In road ecology, the CPSS paradigm may involve a formulation with 1. Computational models processing wildlife movement data, traffic patterns, and environmental conditions (cyber); 2. Road networks, natural habitats, and physical barriers (physical); and 3. Human activities such as land development, transportation policies, and conservation initiatives (social). Integrating geospatial data could allow us to map wildlife migration routes, collision hotspots, and habitat connectivity, while accounting for temporal dynamics, including seasonal migration periods, breeding seasons, and variations in traffic over time.

Through the Evolving CPSS aspect, designers can enable adaptive modeling by updating predictions as new wildlife movement data becomes available or as environmental conditions change, for example, due to climate change. With proactive planning designers may anticipate shifts in wildlife patterns and adjust infrastructure plans accordingly.

For instance, engineers and ecologists could use our approach to identify high-risk areas by combining data on wildlife movements, road locations, and automobile traffic volumes. This informs the design of effective wildlife overpasses or underpasses that align with animal behavior and movement patterns. Continuous post-implementation data collection allows for assessing the effectiveness of these structures and making necessary adjustments.

The benefits of applying our modeling approach in road ecology could include enhanced safety by reducing wildlife-vehicle collisions, conservation goals by maintaining habitat connectivity, and efficient resource use by allocating funds and efforts where they will have the most significant impact.

3.6 Closing Remarks for Chapter 3

In this chapter, we uncover interactions between Artificial Light at Night (ALAN) and bird strikes as a cyber-physical-social system (CPSS), with implications for public policy and aviation safety. Through the development of a predictive model, we have illuminated the impact of ALAN on increasing the incidence of bird strikes, particularly in areas with high levels of light pollution and during critical migration periods. Moreover, we have taken steps to address the previously identified research gaps.

Summary of Results

Our findings confirm that ALAN is a contributing factor to bird strikes, with clear correlations between increased light pollution levels and the frequency of these incidents. Policymakers and conservationists can use the model we develop here to estimate relative strike risk across locations and dates in the United States for which light pollution and migration data exist, and to target mitigation accordingly. By integrating data from the FAA Wildlife Strike Database, a light pollution GIS map, and BirdCast migration forecasts, we have demonstrated the potential of predictive modeling in informing more effective and proactive approaches to aviation safety.

Implications for Public Policy and Lighting Design

From our results, policymakers and lighting designers can act in two ways. By adopting bird-friendly lighting technologies and incorporating ALAN considerations into the planning process,

policymakers can play a critical role in reducing the negative impacts of artificial lighting on avian populations. If city governments set and enforce lighting standards, they can reduce the exposure of birds to artificial light in urban environments. Thus, in summary, we propose an approach for system designers to design public policies through model-based system design by allowing them to:

1. Analyze the interaction between component-level variables and assess their impact on the emergent behavior of the system.
2. Analyze interactions between transdisciplinary variables comprising diverse social and technical domains.
3. Utilize model-based system design for better-informed decision-making.
4. Investigate and discern the primary variables with the most significant impact on bird strikes.

Looking Ahead

With this work, we provide an actionable step for engineering informatics in aviation safety, and more generally towards designing Evolving Cyber-Physical-Social Systems (E-CPSS) for public policy. By adopting an informed, proactive approach to public policy design, we can mitigate the adverse effects of ALAN and foster a safer, more sustainable coexistence between human and avian populations. Further, we envision designing adaptive systems that support self-adaptive public policies, enabling decision-makers to make better-informed decisions pivoted at the integration of preserving wildlife, preventing flight accidents, and informed urban planning for a sustainable tomorrow.

For inquiries about accessing our dataset and using our data to build upon our work, please contact the corresponding author.

We opened this chapter by setting out to address RQ1 and to examine Hypothesis 1. Within the scope of the ALAN–bird strike problem we study, the evidence we present is consistent with that hypothesis: organizing observational features into cyber, physical, and social categories before model construction, and interpreting outputs through SHAP attribution mapped back to those same categories, yields a state characterization that is interpretable as a description of the tri-space system state while remaining comparably accurate, and we identify light pollution level as the strongest predictor of bird strike incidence among the factors we consider. On that basis we offer

the new knowledge we set out to produce: a CPSS-structured predictive modeling method for constructing interpretable tri-space state characterizations of an E-CPSS, together with empirical confirmation, for this dataset, that Artificial Light at Night is a statistically significant contributing factor to bird strikes. We take this to contribute I(si), the first formal component of the DSIM, on which the subsequent layers build. With respect to verification and validation, we read the Frame of Reference as offering theoretical structural validity, the ALAN–bird strike problem as a reasonable vehicle for empirical structural validity, and the reported accuracy and interpretability as empirical performance validity within the bounds of this single problem; whether the method generalizes beyond it, its theoretical performance validity, we defer to the cumulative argument of Chapter 9 and do not claim here.

When designers read the state of an E-CPSS they learn its character, and they do not thereby learn what to do about it. Knowing which factors matter, and in which space they lie, shows a designer where to intervene; it does not tell them which decisions to make there, how those decisions couple, or which regions of the design space they can still accept once they account for uncertainty. That is the work of Chapter 4, where we move from I(si) to D(si), the intra-state decision map, and develop the mathematics for structuring robust decisions within a known state.

References

[1] Haynes, J. P. E., Bhalerao, M. J., Honeycutt, W. T., Allen, J. K., and Mistree, F., 2024, "Predictive Modelling for Public Policy Design: The Impact of Artificial Lights at Night (ALAN) on Bird Strikes," American Society of Mechanical Engineers Digital Collection. https://doi.org/10.1115/DETC2024-143780

[2] Clark, A., 2009, "Plane Crashes in Hudson River in New York," *The Guardian*, accessed March 5, 2024, https://www.theguardian.com/world/2009/jan/15/us-airways-plane-crash-new-york-hudson-river

[3] Longcore, T., and Rich, C., 2004, "Ecological Light Pollution," *Frontiers in Ecology and the Environment*, 2(4), pp. 191–198.

[4] Dolbeer, R. A., 2011, "Increasing Trend of Damaging Bird Strikes With Aircraft Outside the Airport Boundary: Implications for Mitigation Measures," *Human–Wildlife Interactions*, 5(2), pp. 235–248.

[5] Sanders, D., Frago, E., Kehoe, R., Patterson, C., and Gaston, K. J., 2021, "A Meta-Analysis of Biological Impacts of Artificial Light at Night," *Nature Ecology and Evolution*, 5(1), pp. 74–81.

[6] Rodríguez, A., Holmes, N. D., Ryan, P. G., Wilson, K.-J., Faulquier, L., Murillo, Y., Raine, A. F., Penniman, J. F., Neves, V., Rodríguez, B., Negro, J. J., Chiaradia, A., Dann, P., Anderson, T., Metzger, B., Shirai, M., Deppe, L., Wheeler, J., Hodum, P., Gouveia, C., Carmo, V., Carreira, G. P., Delgado-Alburqueque, L., Guerra-Correa, C., Couzi, F.-X., Travers, M., and Corre, M. L., 2017, "Seabird Mortality Induced by Land-Based Artificial Lights," *Conservation Biology*, 31(5), pp. 986–1001.

[7] Bibri, S. E., 2021, "Data-Driven Smart Sustainable Cities of the Future: Urban Computing and Intelligence for Strategic, Short-Term, and Joined-Up Planning," *Computational Urban Science*, 1(1), p. 8. https://doi.org/10.1007/s43762-021-00008-9

[8] Kovacs-Györi, A., Ristea, A., Havas, C., Mehaffy, M., Hochmair, H. H., Resch, B., Juhasz, L., Lehner, A., Ramasubramanian, L., and Blaschke, T., 2020, "Opportunities and Challenges of Geospatial Analysis for Promoting Urban Livability in the Era of Big Data and Machine Learning," *ISPRS International Journal of Geo-Information*, 9(12), p. 752. https://doi.org/10.3390/ijgi9120752

[9] Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F., 2001, "Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," *AIAA Journal*, 39(12), pp. 2233–2241. https://doi.org/10.2514/2.1234

[10] Amini, S., Saber, M., Rabiei-Dastjerdi, H., and Homayouni, S., 2022, "Urban Land Use and Land Cover Change Analysis Using Random Forest Classification of Landsat Time Series," *Remote Sensing*, 14(11), p. 2654. https://doi.org/10.3390/rs14112654

[11] Triantakonstantis, D., and Stathakis, D., 2015, "Urban Growth Prediction in Athens, Greece, Using Artificial Neural Networks," *International Journal of Civil, Environmental, Structural, Construction and Architectural Engineering*, 9(3), pp. 234–238.

[12] Crispim-Mendes, T., Valerio, F., Marques, A. T., Pita, R., Godinho, S., and Silva, J. P., 2024, "High-Resolution Species Distribution Modelling Reveals Spatio-Temporal Variability of Habitat Suitability in a Declining Grassland Bird," *Landscape Ecology*, 39(3), p. 49. https://doi.org/10.1007/s10980-024-01848-6

[13] Bühlmann, P., and van de Geer, S., 2011, "Generalized Linear Models and the Lasso," *Statistics for High-Dimensional Data: Methods, Theory and Applications*, P. Bühlmann and S. van de Geer, eds., Springer, Berlin, Heidelberg, pp. 45–53. https://doi.org/10.1007/978-3-642-20192-9_3

[14] Hoarau, A., Martin, A., Dubois, J.-C., and Le Gall, Y., 2023, "Evidential Random Forests," *Expert Systems With Applications*, 230, p. 120652. https://doi.org/10.1016/j.eswa.2023.120652

[15] Aach, M., Inanc, E., Sarma, R., Riedel, M., and Lintermann, A., 2023, "Large Scale Performance Analysis of Distributed Deep Learning Frameworks for Convolutional Neural Networks," *Journal of Big Data*, 10(1), p. 96. https://doi.org/10.1186/s40537-023-00765-w

[16] Gupta, H., Bhalerao, M. J., Honeycutt, W. T., Allen, J. K., and Mistree, F., 2025, "On Designing Evolving Cyber-Physical-Social Systems (E-CPSS): A Decision-Based Design Perspective," *Journal of Computing and Information Science in Engineering*, pp. 1–86. https://doi.org/10.1115/1.4070094

[17] Wang, F.-Y., 2010, "The Emergence of Intelligent Enterprises: From CPS to CPSS," *IEEE Intelligent Systems*, 25(4), pp. 85–88. https://doi.org/10.1109/MIS.2010.104

[18] Alsamhi, S. H., Kumar, S., Hawbani, A., Shvetsov, A. V., Zhao, L., and Guizani, M., 2024, "Synergy of Human-Centered AI and Cyber-Physical-Social Systems for Enhanced Cognitive Situation Awareness: Applications, Challenges and Opportunities," *Cognitive Computation*, 16(5), pp. 2735–2755. https://doi.org/10.1007/s12559-024-10271-7

[19] Allen, J. K., Nellippallil, A. B., Ming, Z., Milisavljevic-Syed, J., and Mistree, F., 2023, "Designing Evolving Cyber-Physical-Social Systems: Computational Research Opportunities," *Journal of Computing and Information Science in Engineering*, 23(060815). https://doi.org/10.1115/1.4062883

[20] Mariani, S., Picone, M., and Ricci, A., 2022, "About Digital Twins, Agents, and Multiagent Systems: A Cross-Fertilisation Journey," *Autonomous Agents and Multiagent Systems. Best and Visionary Papers*, F. S. Melo and F. Fang, eds., Springer International Publishing, Cham, pp. 114–129. https://doi.org/10.1007/978-3-031-20179-0_8

[21] Kordabad, R. A., 2022, "Managing Computational Complexity Through Using Partitioning, Approximation, and Coordination," Ph.D. dissertation, School of Industrial and Systems Engineering, University of Oklahoma.

[22] Carré, C., and Hamdani, Y., 2021, "Pyramidal Framework: Guidance for the Next Generation of GIS Spatial-Temporal Models," *ISPRS International Journal of Geo-Information*, 10(3), p. 188. https://doi.org/10.3390/ijgi10030188

[23] "FAA Wildlife Strike Database," accessed March 5, 2024, https://wildlife.faa.gov/home

[24] "Wildlife Hazard Mitigation | Federal Aviation Administration," accessed March 5, 2024, https://www.faa.gov/airports/airport_safety/wildlife

[25] "Light Pollution Atlas 2020," accessed March 5, 2024, https://djlorenz.github.io/astronomy/lp2020/

[26] Cinzano, P., Falchi, F., and Elvidge, C. D., 2001, "The First World Atlas of the Artificial Night Sky Brightness," *Monthly Notices of the Royal Astronomical Society*, 328(3), pp. 689–707.

[27] "BirdCast - Bird Migration Forecasts in Real-Time," accessed March 5, 2024, https://birdcast.info/

[28] Van Doren, B. M., and Horton, K. G., 2018, "A Continental System for Forecasting Bird Migration," *Science*, 361(6407), pp. 1115–1118. https://doi.org/10.1126/science.aat7526

[29] Chung, S. Y., Venkatramanan, S., Elzain, H. E., Selvam, S., and Prasanna, M. V., 2019, "Chapter 4 - Supplement of Missing Data in Groundwater-Level Variations of Peak Type Using Geostatistical Methods," *GIS and Geostatistical Techniques for Groundwater Science*, S. Venkatramanan, M. V. Prasanna, and S. Y. Chung, eds., Elsevier, pp. 33–41.

[30] Yao, X. A., 2020, "Georeferencing and Geocoding," *International Encyclopedia of Human Geography (Second Edition)*, A. Kobayashi, ed., Elsevier, Oxford, pp. 111–117.

[31] Dolbeer, R. A., 2006, "Height Distribution of Birds Recorded by Collisions With Civil Aircraft," *The Journal of Wildlife Management*, 70(5), pp. 1345–1350.

[32] Kyba, C. C. M., Kuester, T., Sánchez de Miguel, A., Baugh, K., Jechow, A., Hölker, F., Bennie, J., Elvidge, C. D., Gaston, K. J., and Guanter, L., 2017, "Artificially Lit Surface of Earth at Night Increasing in Radiance and Extent," *Science Advances*, 3(11), p. e1701528.

[33] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É., 2011, "Scikit-Learn: Machine Learning in Python," *The Journal of Machine Learning Research*, 12(85), pp. 2825–2830.

[34] Schlosser, L., Hothorn, T., and Zeileis, A., 2019, "The Power of Unbiased Recursive Partitioning: A Unifying View of CTree, MOB, and GUIDE," arXiv:1906.10179.

[35] Falchi, F., Cinzano, P., Duriscoe, D., Kyba, C. C. M., Elvidge, C. D., Baugh, K., Portnov, B. A., Rybnikova, N. A., and Furgoni, R., 2016, "The New World Atlas of Artificial Night Sky Brightness," *Science Advances*, 2(6), p. e1600377.

[36] Hoepfner, G., Nachmann, I., Zerwas, T., Berroth, J. K., Kohl, J., Guist, C., Rumpe, B., and Jacobs, G., 2023, "Towards a Holistic and Functional Model-Based Design Method for Mechatronic Cyber-Physical Systems," *ASME Journal of Computing and Information Science in Engineering*, 23(051001). https://doi.org/10.1115/1.4056807

[37] Hanke, A., Vernica, T., and Bernstein, W. Z., 2020, "Linking Performance Data and Geospatial Information of Manufacturing Assets Through Standard Representations," American Society of Mechanical Engineers Digital Collection. https://doi.org/10.1115/DETC2020-22300

[38] Pellicciari, M., Andrisano, A. O., Leali, F., and Vergnano, A., 2009, "Engineering Method for Adaptive Manufacturing Systems Design," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 3(2), pp. 81–91. https://doi.org/10.1007/s12008-009-0065-9

[39] Nievas, N., Espinosa-Leal, L., Pagès-Bernaus, A., Abio, A., Echeverria, L., and Bonada, F., 2024, "Offline Reinforcement Learning for Adaptive Control in Manufacturing

Processes: A Press Hardening Case Study," *ASME Journal of Computing and Information Science in Engineering*, 25(011004). https://doi.org/10.1115/1.4066999

[40] Singh, G., and Ahuja, I. S., 2012, "Just-in-Time Manufacturing: Literature Review and Directions," *International Journal of Business Continuity and Risk Management*, 3(1), pp. 57–98. https://doi.org/10.1504/IJBCRM.2012.045519

[41] Tekinbaş, C., Alpaslan, F. N., and Güçlü, A. N., 2025, "Risk Management Based on Machine Learning," *ASME Open Journal of Engineering*, 4. https://doi.org/10.1115/1.4069023

[42] Siers, S. R., Reed, R. N., and Savidge, J. A., 2016, "To Cross or Not to Cross: Modeling Wildlife Road Crossings as a Binary Response Variable With Contextual Predictors," *Ecosphere*, 7(5), p. e01292. https://doi.org/10.1002/ecs2.1292

.

CHAPTER 4

DESIGNING ROBUST POLICIES FOR EVOLVING CYBER-PHYSICAL-SOCIAL SYSTEMS IN UNCERTAIN ENVIRONMENTS

Background and Context

There is a particular kind of frustration that a designer faces after completing a thorough state characterization: the picture is clear, the contributing factors are named, the space is legible, and yet none of that tells us what to do. Through the state of the Agriculture 5.0 E-CPSS, we observe that six decision variables span three levels of abstraction, and that the interactions between government subsidy policy, regional technology deployment, and individual farmer adoption are bidirectional and nonlinear, and that five competing goals cannot be simultaneously maximized. What we have from Chapter 3 is I(si). What we need now is D(si), the structured decision map which designers use to understand which decisions must be made, how they are coupled, and which regions of the design space remain acceptable when uncertainty is realized.

The challenge of structuring those decisions is not merely one of dimensionality. It is one of conceptual organization. Designers intervene in an E-CPSS at three levels at once: long-term strategic framing, medium-term implementation mechanism, and near-term stakeholder engagement. When designers commit to a technology adoption target at the foundational level, they constrain how they can structure subsidies at the operational level, and that constraint in turn shapes what outreach they must fund for social acceptance. Treating these three levels as equivalent variables in a single optimization loses the structural information that makes the intervention designable at all. Before any mathematics is written, we need a principled scheme for classifying decisions by type, by interaction strength, and by level of abstraction. That is what we provide through the Integrated Policy Designing Decision Model.

In this chapter, we develop the IPDDM as a three-dimensional classification framework, formulate the classified decisions as a coupled system of compromise Decision Support Problems, and apply Design Capability Index constraints to quantify robustness explicitly rather than hoping

that nominal feasibility survives the uncertainty the system will actually impose. The governing philosophy is satisficing: we are not searching for a globally optimal solution under idealized conditions. We are searching for a region of solutions that meet all performance thresholds at an acceptable level across the range of stakeholder priority configurations that an E-CPSS is likely to present. This philosophy, grounded in Simon's concept of bounded rationality and in the decision-based design tradition of Mistree and Allen, is particularly appropriate for E-CPSS design because vertex solutions, the traditional output of optimization, are maximally sensitive to precisely the kind of systemic uncertainty that characterizes these problems.

We demonstrate the method on the Agriculture 5.0 test problem, where five policy goals and six design variables span foundational, operational, and engagement levels of decision. Through the IPDDM classification, we reveal that three of the six decision variables interact bidirectionally and must be formulated as coupled cDSPs rather than solved sequentially. Through DCI constraints, we demonstrate that the satisficing solution space we identify remains feasible across thirty-three stakeholder priority scenarios. In this chapter we produce D(si), the intra-state decision map and second formal component of the DSIM, and through it we show that designers can move from a state characterization to a structured, robustness-verified prescription for intervention. What we do not yet address is the temporal dimension. D(si) is a snapshot of one state. E-CPSS do not stand still, and the decisions designed for the present state must be traceable to the outcomes desired at the next. That is the work of Chapter 5.

Preface

Policymakers in complex decision environments face challenges driven by uncertainty, interdependent choices, and evolving conditions. n these environments policymakers must make several interacting decisions at different levels of abstraction, sometimes concurrently and sometimes separately, and the outcome of one decision bears on another. Policymakers who apply traditional optimization with incomplete information obtain vertex solutions, and those solutions change sharply under small changes in their assumptions. In this chapter, we address the need for robust policy design methods embodied in satisficing principles capable of handling such complexities. By modeling policies as Evolving Cyber-Physical-Social Systems, we propose a framework through which designers structure policy decisions under uncertainty and verify robustness explicitly.

To validate our method and the framework, we apply it to a sociotechnical test problem - Agriculture 5.0, addressing modern farming challenges such as technology adoption, resource challenges, and social acceptance. Through this test problem, we assess the effectiveness of the method (and the framework) in designing policies that are robust and balance various levels of abstraction.

In this chapter we specifically address: what methods and principles do designers use to account for uncertainty when they design robust policies for complex systems? Our contributions include: (i) evaluation of the key fundamental literature on robust policy designing; (ii) an integrated decision classification scheme (IPDDM) for structuring decisions at multiple levels of abstraction, (iii) a robust decision-support framework based on coupled decision method, (iv) scenario-driven evaluation of policy trade-offs, and (v) the application of DCI for quantifying robustness in policy designing.

Research Question, Hypothesis, and New Knowledge in this Chapter

Research Question	RQ2a. What is the mathematics and method for designing robust strategies in an E-CPSS through coupled, multi-level decisions under uncertainty?
Hypothesis	Hypothesis 2a. By modeling E-CPSS interventions as coupled compromise Decision Support Problems that encode decision types, interaction strengths, and hierarchical levels using the Integrated Policy Designing Decision Model, with Design Capability Index constraints enforcing robustness explicitly, we produce satisficing solutions that remain robust across stakeholder priority configurations under uncertainty.
New Knowledge	New Knowledge 2a. A coupled decision method anchored in the Integrated Policy Designing Decision Model and Design Capability Indices for designing robust multi-level satisficing strategies in Evolving Cyber-Physical-Social Systems under uncertainty.

Key Words – Policy Design(ing), Uncertainty, Robust, Coupled Decisions, Dynamic Environment, Decision Levels, Decision interactions, Evolution, Cyber-Physical-Social System, Agriculture 5.0

4.1 Frame of Reference

Policy Design(ing) is a branch of policy sciences concerned with the study of policy processes [1]. In policymaking we can often see a clear contrast: some policies are carefully planned and well-constructed, while others seem random or rushed [1]. This raises the question of why some policymaking efforts are thoughtful and logical, while others are not. The focus through the idea of "policy design(ing)" is on creating policies based on evidence, logic, and careful planning to address specific problems effectively. In an ideal world, policymakers would always follow this approach, using facts and proven methods to guide their decisions. However, this is not always how things work in reality. In many cases, policymaking is messy and complicated. Policymakers often face limits on information or knowledge, making it hard to make fully informed decisions. Additionally, policies are rarely made by one person; they involve many stakeholders who may have conflicting goals or compete with one another. These challenges make it difficult to maintain the structured, logical process associated with policy design. As researchers like Schön [2], Gero [3,4] and Hudson and co-authors [5] have pointed out, when collaboration breaks down or information is lacking, policymaking becomes more of a "non-design" process, driven by disagreements or guesswork. Similarly, Howlett and authors [6] argue that policymakers rarely follow a purely logical or evidence-based approach, even though this is often seen as the ideal. Instead, real-world policymaking is shaped by practical constraints, competing interests, and unpredictable circumstances.

In many cases, policy decisions lack meaningful consideration of design principles. Instead, the focus may be overtly partisan or driven by self-interest, with little to no effort to gather evidence about "what works and when" to achieve effective or efficient outcomes for the people affected by these policies [7, 8]. Because policymakers do not design in this sense, we develop a method through which they can. In this chapter, we address this gap by proposing a method to support the development of effective policies. Effective policy designing involves using evidence, analysis, and structured methods to craft solutions that address specific challenges and deliver meaningful results. When policymakers ignore these principles, they may produce policies that cost more than they deliver, or that harm the people they intend to help. Developing systematic approaches to designing policy can help counter these issues. By emphasizing evidence-based practices, stakeholder engagement, and a focus on clear goals, policymakers can create strategies that are

more aligned with public needs and interests [9]. Without such methods, policy deliberations are likely to remain fragmented and driven by short-term interests, failing to deliver long-term benefits to society. Policymaking is inherently uncertain, even under ideal conditions. The modern study of policy has long recognized that public policies often emerge from the exercise of power rather than from the application of knowledge. Even in cases where knowledge takes center stage, it does not always ensure the successful passage, implementation, or achievement of desired outcomes [10-13]

Governments, even with the best intentions, frequently face complex challenges [1]. These challenges are compounded by the involvement of numerous stakeholders, diverse ideas, and competing interests [14]. Policymaking is often carried out in dynamic environments where problems evolve over time, making it hard to build or sustain consensus on the most viable policy options. This constant flux complicates the decision-making process, leaving policymakers struggling to identify and implement strategies that can address the issues effectively. The interplay of these factors underlines the difficulty of designing and executing policies in a way that consistently delivers intended results. This also highlights the importance of robust and flexible approaches to policymaking that can respond to changing circumstances and the intricate nature of modern governance.

To address the intricacies in designing policies, we posit that designing policy can be modeled as evolving cyber physical social systems (E-CPSS) [15,16, 90]. Designing policy as an E-CPSS treats policymaking as a dynamic and integrated process. “Cyber” refers to real-time data and digital networks, which help policymakers monitor unfolding conditions. “Physical” covers the tangible elements such as infrastructure, equipment, and resource flows that shift with technological innovation and environmental factors. In social entity the focus is on human behavior and community engagement, ensuring people’s values and actions shape (and reshape) policy outcomes. By accounting for these three interconnected dimensions, policymakers recognize that each decision can trigger changes across the entire system. hrough an E-CPSS approach, policymakers design adaptively rather than issue static directives, and they respond to new information, shifting stakeholder needs, and emerging technologies. It is widely accepted in the field of policy studies that well-designed policies are more likely to succeed in addressing the issues they are intended to solve. Conversely, policies that are poorly designed or lack intentional

design often fail due to their inability to effectively identify or tackle the core problems. Policy scholars therefore treat systematic policy development as central [1, 2, 17]. The foundation of modern policy sciences rests on the belief that accumulated knowledge about policy tools, their effects, and their impacts, gained through years of governance and state-building experience, can significantly improve the ability to mobilize resources and achieve meaningful outcomes. Through such approaches designers aim to enhance public value by leveraging tried-and-tested policy mechanisms [1, 18, 19].

However, significant gaps remain in understanding the processes that designers can use to design effective policies when viewed from holistic systems perspective. Questions persist about how the selection of specific policy tools influences their success, and why some choices result in better outcomes than others [1,2]. Due to uncertainties there is a need for continued exploration of how policies can be designed more effectively to meet societal needs and deliver tangible benefits. The inference is clear; investing in the study of policy designing processes to design robust policies and their impacts is critical to improving governance and achieving desired results. In this chapter, this challenge has been closely examined by proposing a method for designing robust policies that accounts for uncertainties and addresses the diverse characteristics inherent in policy designing challenges, as discussed throughout this work.

In Section 4.1, we establish the frame of reference. In Section 4.2, we review how researchers address robustness and uncertainty in policy design. In Section 4.3, we introduce Agriculture 5.0 as the test problem. In Section 4.4, we develop the proposed framework and the Coupled Decision Method through the Integrated Policy Designing Decision Model. In Section 4.5, we explain how designers use the Design Capability Index to quantify robustness. In Section 4.6, we demonstrate the method through mathematical formulation, scenario analysis, and visualization of satisficing policy options. In Section 4.7, we provide our closing remarks.

4.2 Foundations for Effective Policy Designing

In the state of that art, we can see some confusion in distinguishing between the definitions of policymaking and policy design(ing) [20]. Designing policy involves a deliberate and intentional process of defining objectives and strategically linking them to policy tools aimed at achieving those objectives [20-24]. Unlike general policymaking, policy designing represents a specific

approach to policy formulation, one that emphasizes structured activities such as gathering knowledge about the impacts of various policy instruments on their intended targets and evaluating the relevance of this knowledge to designing effective and implementable policies. Thus, through policy designing, designers can bridge the gap between theoretical aspirations and practical implementation by incorporating evidence-based methods and systematic analysis[20,25,28]. Through policy designing, policymakers develop targeted policies and can trace each instrument to the objective it serves. Davis Boborw [25] states that policy design is ubiquitous, necessary and difficult' but it is less studied and understood. Linder and Peters [29] have tied the policy and design by referring it to 'spatial' activity where possible and feasible design choices exist within a defined policy 'space' – In exact words *"a systematic activity composed of a series of choices . . . design solutions, then, will correspond to a set of possible locations in a design space . . . this construction emphasizes not only the potential for generating new mixtures of conventional solutions, but also the importance of giving careful attention to trade-offs among design criteria when considering instrument choices"*

In the method proposed in this chapter, we draw on this *quoted* idea as a foundational principle, emphasizing its significance for designing robust policies. e use this idea as a lens through which designers address uncertainty and complexity, and we tie our framework to it. One of the central themes in the field of policy design(ing) is the creation of effective and impactful policies. Howlett [20] highlights that policy effectiveness is an 'elusive' goal which is difficult to achieve. The author highlights two primary reasons for the difficulty in designing policies: first, the uncertainties that policymakers encounter during the design process, and second, the potential for errors, compounded by the fact that some policies may emerge from non-design processes. This observation is crucial as it not only underscores the need to address uncertainties but also emphasizes the importance of developing design methods through which policy designers can incorporate these complexities into the policy design process and subsequently model robustness through appropriate metrics. Through the approach introduced in the chapter we accord with this observation. Establishing clear definitions for policy design(ing) is essential, particularly for designers and researchers working on developing robust methods for designing effective policies. In this chapter, we ascribe to the following foundational idea as ascribed by Linder and Peters [29] –

"Whether the problem is an architectural, mechanical, or administrative one, the logic of design is fundamentally similar. The idea is to fashion an instrument that will work in a desired manner. In the context of policy problems, design involves both a systematic process for generating basic strategies and a framework for comparing them. Examining problems from a design perspective offers a more productive way of organizing our thinking and analytical efforts."

Through designing as a systematic and purposeful process, designers can bridges disciplines and provide a universal logic for addressing complex policy challenges. In policy designing, the emphasis is on both strategy generation and evaluation within a coherent framework, helping manage uncertainties and trade-offs while effectively configuring policy tools for desired outcomes. In Vincent Ostrom's article [30], *The Intellectual Crisis in American Public Administration*, this connection is further reinforced by emphasizing the critical role of design in understanding and improving system performance - "*An appropriate theory of design is necessary both to understand how a system will work and how modifications or changes in a system will affect its performance*" Ostrom's highlights the intersection of uncertainty and design, a central theme in this article and the proposed method. Policy makers operate within complex and ever-changing systems, where uncertainties ranging from unforeseen societal behaviors to external shocks must be accounted for to ensure effective performance. Robustness becomes a key consideration in this context, through which we address the need for policies to function reliably despite these uncertainties. By embedding robustness into the design process, policymakers can create systems capable of sustaining performance over time. This is further expanded in Section 4.2.1.

4.2.1 Uncertainty, Robustness, and Policy Designing

The primary goal in policy design(ing), and by extension policymakers, is to create solutions (policy options) that address collective problems in a sustainable and enduring manner. Policy decisions should not only address immediate concerns but also remain relevant and effective over time [31]. Durability in policymaking involves maintaining consistent goals while ensuring that the mechanisms in place continue to function at an acceptable level of effectiveness in achieving these objectives. As pointed in [32-34] policy designers should have the ability to incorporate variability to account for uncertainties while ensuring that policies remain effective to changing

contexts and challenges. This aspect has been discussed in detail in Section 4.6 where we have detailed about incorporating robustness in policy designing.

The emphasis on robustness underscores the importance of thoughtful policy designing. By incorporating principles from design theory such as adaptability, feedback mechanisms, and stakeholder engagement policies can be structured to remain functional and effective over time. Policymakers must engage in ongoing evaluation and make adjustments to ensure policies adapt to emerging challenges and shifting contexts. Thus, without a deliberate focus on designing robust solutions, policies risk becoming obsolete or ineffective, ultimately failing to address the collective problems they were designed to solve. To accomplish this, policy designers and scientists must incorporate policy robustness. As Capano and Woo define, robustness can be defined as the "capacity of policymaking to respond to and retain functionality amid uncertainty" [31].

Decision-making often involves managing unavoidable uncertainty [36-38]. Because researchers acknowledge uncertainty as part of long-term decision processes [34], they have developed tools and strategies for it (refer to Walker and co-authors [39] for a comprehensive review). Decisions in policy designing exist at various levels, and considering these are critical for developing robust policies, as further elaborated in Section 4.4. Robust decision-making represents a leading example of modern, model-based decision support frameworks designed to address deep uncertainty [37,40,41]. This approach involves generating an extensive range of plausible future scenarios that encompass key uncertainties [42]. These scenarios are a testing ground for evaluating proposed policies, providing insights into their performance across various conditions. Through a process known as scenario discovery, the critical vulnerabilities and potential opportunities of a given policy are identified [37, 43, 44]. Understanding uncertainty and robustness in the context of policy designing is essential. In the following sub-sections we delve deeper into these concepts and explore how they can be integrated into policy frameworks to enhance effectiveness of designing policy.

4.2.1.1 Uncertainties in Policy Designing

As pointed in Section 4.2.1, uncertainty is an inherent aspect of policy design, stemming from a variety of sources, including the complexity of societal challenges, the dynamic nature of policy environments, and the unpredictability of human behavior. Policymakers must often make decisions without complete information, relying on assumptions and projections that may not fully

capture future realities [45]. This inherent uncertainty is compounded by statistical uncertainties and inaccuracies in data, as well as the imprecision in estimates concerning the expected relationships between policy interventions and the behaviors they aim to influence [46]. These layers of uncertainty present significant challenges for policy designing, as they can affect both the formulation of policies and the likelihood of achieving desired outcomes. Designers must recognize and address these uncertainties if they are to produce policies that keep performing as conditions change. While designing policies there is a significant challenge due to the need to acknowledge, comprehend, and navigate uncertainty. This challenge arises because [39]:

- It is impossible to eliminate all uncertainties about the future.
- Overlooking uncertainty may restrict the ability to implement corrective measures when needed, potentially leading to avoidable adverse outcomes.
- Neglecting uncertainty can also result in missed opportunities, ultimately compromising the sustainability of the plans.

Uncertainties linked to the temporal dimension of policy designing that brings variability play a crucial role in determining a policy's long-term success. If the policies designed through model based policy designing are not robust, they are bound to render inadequate. One of the aspects to address this is incorporating variability in the goals that are achieved through the policy designing. An important yet often overlooked aspect of policy designing is its temporal nature [46]. Stakeholders expect a policy to work on implementation and to keep working [20, 46, 47]. However, the mechanisms through which this long-term effectiveness can be achieved remain insufficiently understood. Traditional scientific approaches in fields such as engineering, social sciences, and natural sciences have long assumed that uncertainties arise primarily from either a lack of information or random variation [37]. Through this assumption, the efforts are driven to reduce uncertainty by focusing on extensive information gathering and processing [39, 48] or by addressing variability through stochastic processes and statistical analyses. While these methods are effective in many domains, there are limitations in addressing the types of uncertainties that characterize today's strategic planning challenges. Most contemporary strategic planning problems involve uncertainties about the future that cannot be resolved by merely acquiring more information or applying statistical models [49]. These uncertainties are often systemic and deeply rooted in the unpredictability of future events, making them fundamentally different from traditional notions of randomness or information gaps. Thus, there is a critical gap in policy

designing theory and practice, calling for further exploration of how policies can be structured to endure and adapt to future challenges while sustaining their intended outcomes. For policies, even those that are relatively straightforward, to remain effective over time amid evolving challenges, shifting environments, and ongoing policy feedback, they must possess a baseline level of robustness [46]. Through this robustness, we can ensure that policies can endure disruptions and maintain their core functions and identity [50].

As Howlett and colleagues [20] point out, uncertainty in policy designing often leads to two common pitfalls: under-design and over-design. Under-design occurs when policymakers underestimate or underreact to a problem, failing to address it adequately. Conversely, over-design arises when a problem is overestimated, prompting an excessive or disproportionate response. Both scenarios highlight the challenges uncertainty poses in striking the right balance in policy formulation and implementation.

Policy analysts and strategic planners often acknowledge the presence of uncertainty but frequently operate as though the future can be accurately predicted. They tend to create static plans, either an "optimal" plan based on a single "most likely" scenario derived from trend extrapolation or a "robust" plan designed to perform acceptably in a limited set of hypothetical futures [39, 51, 52]. However, if the actual future deviates from these assumptions, these plans are prone to failure. As is the case often, due incomplete information in policy designing the use of optimization methods by getting a vertex solution is inappropriate [53]. McInerney and co-authors [54] vividly describe this approach as akin to "dancing on the tip of a needle," (vertex solutions) emphasizing its precariousness. Designers face greater difficulty because people, institutions, technologies, and natural environments change continually. Yet, despite this dynamic reality, many plans are static and lack provisions for adaptation to new or unforeseen circumstances. Owing to this rigidity the long-term viability is often undermined. By failing to account for this need, policy makers risk creating strategies that quickly become outdated and ineffective, leaving decision-makers unprepared for the complexities of an unpredictable future. When designing policies, decision-makers frequently rely on their existing knowledge, empirical evidence, and external guidance to enhance their capacity to design policies that are both efficient and effective [25, 31, 55, 56]. Designers therefore need to understand robustness in the context of policy design and uncertainty. In Section 4.2.1.2, we examine this relationship.

4.2.1.2 Understanding Robustness in the context of Policy Designing and Uncertainty

Robustness, more frequently examined in the natural sciences than the social sciences, refers to a system's capacity to sustain its core functions or outputs when exposed to shocks or disturbances [31, 57, 58]. Ostrom and co-authors [59] define it as "reduced sensitivity of outputs to shocks," indicating that even if inputs change, the system continues operating effectively. As Capano and Woo [31] emphasize drawing on [60–62] "robustness directly refers to this capacity to deal with external shocks, conditions of uncertainty or imperfect external conditions in a way through which both institutional or procedural functions and the capacity of policy initiatives to attain expected goals are maintained."

In the realm of policy designing, robustness is central to ensuring that policies adapt to unforeseen events yet still deliver their intended results. Uncertainty arises from factors such as limited data, complex political environments, and shifting social landscapes, all of which demand flexible policy architectures. Designers of robust policies include contingency planning, redundancy, and stakeholder engagement, and they build in the ability to recalibrate without abandoning their objectives. They are designed with adaptability in mind, allowing for recalibration without compromising overarching objectives. Designers of robust policies include contingency planning, redundancy, and stakeholder engagement, and they build in the ability to recalibrate without abandoning their objectives.

Policy designing itself has a dual nature, encompassing both the *process* of creating policies and the *products* or instruments that emerge. This distinction between design as a *process* ("verb") and design as a *product* ("noun") parallels the conceptual separation of design as an action versus an outcome, as noted by May [24]. Recognizing these two dimensions provides both scholars and practitioners with a framework to examine and enhance the understanding and practice of policy designing. The process side entails stakeholder negotiation, iterative assessment, and adjustments during implementation, while the product side provides the tangible framework guiding policy action. Through this duality, the importance of robustness across both dimensions becomes evident: a robust design *process* helps manage uncertainty and navigate conflicting interests, and a robust policy *outcome* sustains its functionality despite external shocks.

4.2.1.3 Designing Policy Robustness for Uncertain Environments

Through robust policy designing the aim is to maintain intended policy functions under changing circumstances [31]. While examples of robustness can be seen in technical domains such as inflation-targeting policies and "automatic stabilizers" (progressive taxation, welfare benefits) in macroeconomics [31] their wider application to more socially complex areas remain limited. These financial instruments showcase how specific mechanisms can be structured to adapt to economic fluctuations, but methods for designing robustness in human-centric policy domains require deeper exploration, given the unpredictable nature of social behaviors and values. Designing robust policies is challenging because it calls for solid causal theories and the ability to structure solutions for reliability. Moreover, the *process* of policy design itself must be robust. Designers who attend only to robust outputs may neglect how they can strengthen a policy through iterative refinement during formulation and implementation. In many technically oriented fields, formal models can simulate various scenarios to evaluate policy alternatives systematically [63–65]. However, such modeling is less applicable to socially contested problems without clear definitions or shared objectives.

To obtain robust outputs, designers embed adaptive features in the policy process: stress tests, scenario analysis, and feedback loops [1, 31, 39]. In turn, iterative assessments allow policymakers to recalibrate as external conditions shift. For instance, real-time adjustments during implementation can strengthen a policy's ability to withstand shocks. Ultimately, robustness is an ongoing, dynamic characteristic: policymakers must remain engaged throughout the policy lifecycle to ensure alignment with evolving contexts and stakeholder needs. From both theoretical and practical standpoints, establishing methods for robust policy design is imperative in an era marked by climate change, technological disruptions, and shifting political climates. Capano and Woo [31] call for stronger connections between public policy, institutional analysis, public management, and policy design research, suggesting the development of integrated frameworks that operationalize robustness across diverse contexts. By bridging these fields, policymakers can systematically embed robustness into processes and outcomes, leading to governance structures capable of thriving amid uncertainty.

4.2.2 Towards Multi level Decision Environment for Policy Designing

Uncertainties in policy designing are further magnified in environments where decisions span multiple levels, such as foundational, strategic, and operational layers. These interconnected levels have complex dynamics, as choices at one level often influence or constrain decisions at others. Foundational decisions, which set the long-term vision and objectives of a policy, are typically made with limited information about how future conditions might evolve, introducing uncertainty about their relevance or feasibility over time. Strategic decisions, focused on translating foundational goals into actionable frameworks, must navigate the inherent unpredictability of implementation contexts, including stakeholder dynamics and resource availability. Operational decisions, responsible for day-to-day execution, face uncertainties stemming from immediate challenges and local variations that may not have been fully anticipated during the foundational and strategic phases.

Because designers decide at three levels, uncertainty at one level carries into the others. Policymakers may set foundational goals that do not match conditions on the ground, and those who execute may be unable to adapt as strategic priorities shift. For instance, foundational decisions must be flexible enough to accommodate new information, strategic decisions need to account for evolving conditions and priorities, and operational decisions must integrate feedback loops to address real-time challenges effectively. By embedding robustness across all levels, policymakers can create systems that are not only aligned but also capable of sustaining their intended functions despite the uncertainties inherent in a multi-level decision environment.

Key Insights and Identified Requirements for Developing Robust Policy Designing Methods

In and 4.2 we identify what researchers have established about policy design and what they have not, and we conclude that designers need a structured approach. Researchers have not yet applied design principles to policy in this way, and designers need methods through which they represent interdependence among decisions, quantify interaction strength, and manage uncertainty. Through our findings we emphasize the need for practical frameworks that reconcile trade-offs among efficiency, equity, and sustainability. We advocate for methods that bridge theory and practice, enabling policies to adapt to dynamic challenges by integrating stakeholder goals with operational realities, ultimately yielding coherent, impactful, and robust policy outcomes. Building on the fundamental literature review we address the following research gap in this chapter: *What methods*

and principles effectively account for uncertainty in designing robust policies for complex systems?

To contextualize our proposed method within existing literature, we conducted a comparative analysis of recent key policy design approaches under uncertainty. In Table 4.1 we summarize key methodical features across selected methods, highlighting the comprehensive nature of our approach in addressing multi-level decision-making, interdependencies, robustness, and integration of cyber-physical-social systems.

Table 4.1: Key Literature Review: Comparing Policy Designing Frameworks

Paper	**Models Policy Decisions**	**Uncertainty Management**	**Multi-level / Interdependent Decisions**	**Quantifies Robustness**	**Satisficing Approach**	**Scenario-Based Modeling**
Amorocho-Daza [91]	*	*				*
Athanasoglou [92]	*			*	*	
Liu [93]	*		*			
He [94]	*	*		*		
Motlaghzadeh [95]	*	*	*			*
Ganji [96]	*	*				*
Cai [97]	*	*		*	*	*
Deak [98]	*	*		*		
Fang [99]	*	Partial	*			
Moallemi [100]	*	*		*	*	*
Tasic et al. [101]	*	*	*			*

Castrejon-Campos [102]	*	*	*	*	*	*
Tasic [103]	*		*			

4.3 TEST PROBLEM FOR VALIDATION: AGRICULTURE 5.0

To validate the proposed method, we apply it to a test problem inspired by Agriculture 5.0. In Agriculture 5.0 the convergence of advanced technologies such as IoT, autonomous machinery, and data-driven farming practices to address critical challenges in food security, environmental sustainability, and economic viability is observed. The purpose through this test problem is to demonstrate the utility of the framework to address the complexities and interdependencies inherent in designing policies that encourage the adoption of these technologies while balancing competing objectives and managing uncertainties.

Designing policies for Agriculture 5.0 can be modelled as an Evolving Cyber-Physical-Social System that continuously adapts to shifting technological, environmental, and social dynamics. As agricultural practices become increasingly digitalized, the integration of cyber elements such as data analytics and IoT devices with physical infrastructures like machinery and irrigation systems, and the active participation of farmers and communities, is crucial. This dynamic interplay demands policies that are both robust and flexible, capable of addressing uncertainties and accommodating rapid technological advancements while meeting diverse stakeholder needs.

4.3.1 Significance of Agriculture 5.0 for Policy Designing

Agriculture 5.0 is pivotal in transforming traditional farming into a more sustainable, efficient, and resilient sector. The integration of technology enables precision farming, balanced resource utilization, and reduced environmental impact. However, the adoption of Agriculture 5.0 technologies requires supportive policies that address barriers such as high initial costs, limited

technical knowledge among farmers, and varying levels of readiness across regions. Designing policy in this context is particularly challenging due to:

- **Diverse Stakeholders**: Farmers, governments, technology providers, and environmental organizations, each with different priorities and constraints.
- **Competing Objectives**: Balancing economic incentives, social equity, and environmental sustainability.
- **Uncertainty**: Variability in stakeholder behavior, resource availability, and technology adoption rates.

The goals through this test problems are presented in Section 4.6.

4.3.2 Importance of the Selected Test Problem with Respect to Policy Designing

Designing effective policies requires a structured approach that aligns diverse objectives while addressing the dynamic challenges of implementation. Agriculture 5.0, with its reliance on advanced technologies such as IoT and shared machinery, highlights the critical need for policies that foster inclusivity, promote sustainability, and adapt to stakeholder priorities. The emphasis through policy designing principles is on creating frameworks that balance economic efficiency, social equity, and environmental goals. These principles are essential for developing strategies that cater to the varying needs of stakeholders, ensuring that technological advancements are accessible and beneficial to all.

A key requirement in designing policy is the development of robust policies that perform well under uncertainty. Uncertainties such as fluctuations in adoption rates, regional differences in readiness, and unforeseen resource constraints necessitate policies that can adapt dynamically while maintaining their core objectives. Robust policies mitigate risks by incorporating variability into their planning, ensuring robustness and long-term effectiveness. This approach is particularly vital in Agriculture 5.0, where the pace of technological advancement and diverse regional contexts require adaptable and future-proof strategies. Designing policy for Agriculture 5.0 requires a multi-faceted approach that balances economic efficiency with social equity. Foundational decisions such as subsidy allocation and IoT deployment are directly linked to operational strategies like training programs and stakeholder engagement.

4.3.3 Assumptions

In this chapter, our goal is not to advance Agriculture 5.0 or propose specific policies, but to demonstrate the utility of our method using the proposed test problem. We incorporate realistic assumptions to simplify validation: IoT hubs and shared machinery are scalable based on regional demand, and training programs effectively boost farmer proficiency. A fixed budget for subsidies, IoT deployment, training, and campaigns necessitates trade-offs. Farmer willingness to adopt technologies varies by region, influenced by awareness and financial support. Additionally, differences in rural and urban infrastructure impact deployment priorities and training needs, while policies are evaluated for their long-term effects on reducing resource wastage and promoting sustainable practices.

4.4 PROPOSED FRAMEWORK AND METHOD

In this section we explain the proposed framework and the method. Here are the particular details to understand what is referred as framework and method.

Framework - The framework proposed is a holistic policy designing approach through which designers will begin with identifying the policy problem, structuring decisions through the Integrated Policy Designing Decision Model (IPDDM), applying coupled Compromised Decision Support Problems (c-cDSP) for robust modeling, and leveraging Scenario Modeling and Interpretable Self-Organizing Maps (ISOM) for visualization and interpretation of policy options. By using the framework designers can have a comprehensive pathway from problem definition to actionable policy designing.

Method - The method embedded within this framework integrates the IPDDM classification scheme for structuring decisions and interactions, with coupled Compromised Decision Support Problems (c-cDSP) for mathematical modeling. This combined approach enables the exploration of trade-offs, balance of interconnected decisions, and the development of robust and adaptive policy solutions.

Model – The model in this chapter is referred to the Integrated Policy Designing Decision Model (IPDDM) as a decision classification scheme.

4.4.1 Detailed Description of the Framework

The proposed framework is presented in Figure 4.1. Through this framework, policymakers and designers formulate robust policies while they account for uncertainty and variability in complex decision situations. We incorporate a systematic classification of decisions based on their strengths, types, and interdependencies through the Integrated Policy Designing Decision Model (IPDDM), ensuring a structured foundation for policy formulation. By leveraging coupled Decision Support Problem (cDSP) construct and advanced visualization tools such as Interpretable Self-Organizing Maps (ISOM), through the framework policy designers and decision-makers can explore trade-offs, balance solutions, and interpret patterns within the design solution space. This ensures that policy outcomes are both robust and well-informed. The details of the framework are as follows.

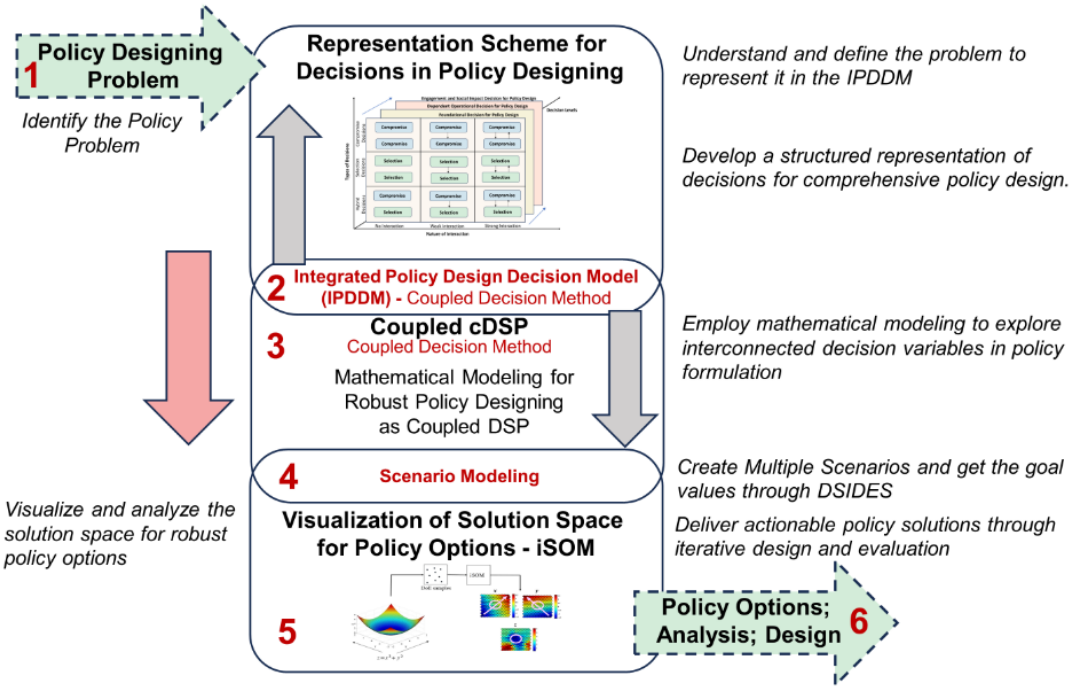


Figure 4.1: Proposed Framework to Designing Policy in Uncertain Environments

Step1 Figure 4.1 - Policy Designing Problem (Refer Figure 4.1)

Description: This stage involves identifying and defining the policy problem. It includes analyzing the scope, stakeholders, and potential impacts of the issue to ensure clarity in problem representation.

Outcome: A clearly defined and structured policy problem.
Impact: A well-defined problem ensures that the subsequent steps address the right objectives and constraints, minimizing ambiguity in policy designing.

Step2 Figure 4.1 - Integrated Policy Designing Decision Model (IPDDM) (Refer Figure 4.1)

Description: Through IPDDM designers will have a structured representation scheme for decisions. IPDDM is an integral component of the coupled decision method. By employing IPDDMS designers can categorize decisions based on levels of engagement, social impact, and the nature of interactions, enabling a comprehensive understanding of the decision-making landscape.
Outcome: A structured decision classification scheme that maps the relationships between critical factors influencing the policy.
Impact: Through this representation we ensure that no key aspect of the problem is overlooked, allowing policymakers to approach the design process systematically and holistically.
IPDDM is explained in detail in Section 4.4.3.

Step3 Figure 4.1 Coupled cDSP (Compromise Decision Support Problems) (Refer Figure 4.1)

Description: Mathematical modeling is applied to explore satisficing design space through trade-offs, and constraints. Coupled cDSP is the mathematical construct for formulating the policy designing problem by using the Coupled Decision Method.
Outcome: A mathematical model that identifies interconnected decisions, balance trade-offs and feasible solutions for policy goals.
Impact: Provides a robust analytical foundation for addressing multi-goal challenges, ensuring policy solutions are robust to conflicting requirements and adaptable to changing priorities.

Step4 Figure 4.1 Scenario Modeling (Refer Figure 4.1)

Description: In Scenario modeling we use DSIDES to attain the values of goals and exercise the cDSP for each goal by generating multiple scenarios with varying wights for goals. It identifies how changes in weights affect outcomes and evaluates a range of possible solutions.
Outcome: A set of well-defined scenarios that highlight the range of potential outcomes and their associated risks.
Impact: By anticipating various scenarios, policymakers are equipped to make informed decisions that account for uncertainty and variability, leading to robust policy designs.

Step5 Figure 4.1* Interpretable Self-Organizing Maps (ISOM) *(Refer Figure 4.1)

Description: Through iSOM designers can visualize and interpret high-dimensional data by clustering and mapping relationships. These insights enable policymakers to understand patterns and trade-offs between policy options.

Outcome: Intuitive visualizations of the solution space that highlight clusters and relationships among policy options.

Impact: Makes complex data interpretable for decision-makers, facilitating evidence-based policy decisions that are grounded in data insights and clear trade-offs.

Step6 Figure 4.1 Policy Options; Analysis; Design (Refer Figure 4.1)

Description: In this stage we well integrate insights from all previous steps to deliver actionable policy recommendations.

Outcome: A set of actionable, evidence-based, and robust policy solutions ready for implementation.

Impact: Empowers policymakers with well-analyzed, adaptable, and effective solutions, ensuring the policy outcomes align with societal goals and stakeholder needs and are robust to uncertainty.

Before delving into the proposed method, it is essential to outline pertinent background that highlight the theoretical foundation and the practical need for such a method. Through this discussion in Section 4.4.2, we establish the relevance of integrating structured decision classification and model based policy designing emphasizing their significance in addressing the complexities, uncertainties, and interdependencies inherent in policy designing.

4.4.2 Understanding the Roots, Concepts, and Ideation

A significant limitation in much of the existing literature on policy instrument choices as explained by [7] is the failure to recognize the inherently multi-level and interconnected nature of policies. Owing to this oversight there is considerable theoretical and terminological confusion [66-68]. Public policies are composed of diverse goals and means that are operational at varying levels of abstraction and application. Consequently, policies are multifaceted, with some components more amenable to deliberate design than others [69].

Understanding policies as multi-level, nested phenomena provide critical insights into the challenges of policy designing. Each level of policy—ranging from foundational objectives to

operational mechanisms—is influenced by different factors and constraints, often creating tension as policymakers attempt to align goals with appropriate instruments. Policy designs(ing) are not merely standalone efforts; they are constrained and shaped by the need to balance expectations and match objectives both within and across categories of policy elements [7, 70, 71]. This interdependence means that decisions made at one level—whether strategic, operational, or foundational often co-determine outcomes at other levels, creating a nested structure that adds complexity to the policy designing process.

In multi-level policy designing the importance of recognizing these embedded relationships to effectively address the challenges of aligning goals and instruments is highlighted. For example, foundational decisions, such as setting long-term policy goals, must account for how these goals will translate into strategic frameworks and operational mechanisms. Many Scholars [7,71 ,72] have noted how these interdependencies restrict the range of viable instrument choices, while Kooiman [73] emphasizes the importance of understanding these nested structures in achieving effective policy outcomes.

A multi-level perspective also underscores the need for adaptive and iterative policy processes. Since policies are implemented and operate in dynamic environments, the relationships between different levels must be continuously monitored and adjusted to respond to changing conditions. For instance, strategic decisions must be flexible enough to adapt operational mechanisms if foundational goals evolve over time. Moreover, this perspective explains why designing policy often encounters practical challenges. Aligning objectives across levels while navigating constraints imposed by existing institutional structures, political dynamics, and resource limitations requires a nuanced approach. Multi-level policy designing frameworks, which consider the interplay between different components, provide a more comprehensive tool for addressing these complexities and crafting policies that are both coherent and resilient. In this chapter we consider different layers of abstractions to model robust policy designing

In this chapter, we have proposed and demonstrated the utility of Interpretable Self organizing Maps (iSOM) in designing policies in interdependent and uncertain environments considering different levels of the system. For details on SOM and iSOM readers are directed to [81-86]. By leveraging iSOM's advanced visualization capabilities, policymakers and designers can explore design solution spaces with multiple conflicting objectives and complex interdependencies.

Moreover, through iSOM the visualization of relationships between variables at different levels of abstraction (86, 88, 89) can be done. Owing to these features iSOM is a robust decision-support framework for addressing the challenges of uncertainty and interdependence in real-world systems, enabling the development of effective, data-driven policies.

4.4.2.1 Considerations for Model Based Policy Designing and Robustness and Uncertainties

In model-based policy designing, designers often rely on models that are abstractions of reality. These models, while useful, are inherently incomplete and incorporate varying levels of fidelity. Traditional approaches frequently focus on static “optimal” solutions, which assume a "most likely" future extrapolated from trends, or static "robust" plans designed to perform acceptably in a limited set of hypothesized scenarios [38, 52]. These predictive methods aim to identify a single best solution. However, under conditions of uncertainty, such approaches falter, as an optimal solution rarely exists, making predictive models unsuitable for effective decision-making [74,75].

This reliance on static approaches becomes even more problematic in environments characterized by numerous interacting decisions, where the cumulative effects of model inaccuracies or incompleteness can result in infeasible or ineffective outcomes. Choi and co-authors, [76] and Allen and co-authors [77] have pointed out that models often fail to account for uncertainties stemming from noise factors (uncontrollable parameters), decision variables (controllable parameters), or the approximations used in creating simulation models. These challenges are further amplified by uncertainty propagation across varying temporal and contextual scales, making it difficult to rely solely on static models. Moreover, empirical data used to inform these models often carries its own uncertainties due to dependencies on environmental conditions, methodologies, and equipment. This highlights the critical need for robust uncertainty management methods in policy designing, as suggested by Seepersad and co-authors [78]. Effective approaches must address these uncertainties while ensuring that policies remain robust in the face of incomplete or imperfect information.

There should be a shift in robust policy designing principles from static, predictive methods and instead focus on solutions that remain effective across a range of scenarios and uncertainties. This includes identifying multiple satisficing solutions that meet acceptable performance thresholds rather than relying on narrowly defined "optimal". By integrating iterative and adaptive processes, policies can be adjusted dynamically to respond to evolving conditions and unforeseen

challenges. Such principles align with robust design methodologies originally developed in engineering and now provide a valuable framework for designing policy. To address these challenges, robust policy designing principles must be integrated into the modeling and decision-making process. By incorporating robust design principles, policymakers can create solutions that are less sensitive to inaccuracies and more adaptable to dynamic environments. This approach not only enhances the robustness of individual policy decisions but also strengthens the overall framework, ensuring that policies remain effective and aligned with their objectives despite the uncertainties inherent in the policymaking process.

While considering another critical perspective in the proposed method is modelling robustness. In this article, a method for achieving robust policy designing in multi-level and coupled decision environments is developed using principles inspired by the Decision Support Problem Technique (DSPT). Originally proposed by Mistree and collaborators and used for various engineering applications [76-80] , DSPT offers a valuable method that can be adapted to policy designing, providing strategies to create resilient and adaptive policies capable of addressing uncertainties and complexities in interconnected systems. When applied to policy designing, the framework facilitates structured decision-making by acknowledging interdependencies, managing uncertainties, and fostering robust solutions that align with long-term objectives. Several critical features of this approach are particularly relevant to the policy domain. A robust policy designing framework begins by partitioning policies into interrelated decisions, recognizing that policies comprise interdependent decisions operating across foundational, strategic, and operational levels. Adapted from DSPT principles, this partitioning enables policymakers to break down complex policy systems into manageable components based on decision types, interactions, and levels, aligning overarching goals with specific tools while ensuring coherence. It further emphasizes evaluating alternative policy pathways, similar to engineering design, by systematically exploring a wide range of interventions that are flexible and adaptable amid uncertainty. Recognizing the challenge of managing uncertainty in policy interactions, the framework incorporates tools to handle the unpredictable influence among decisions across different levels. Finally, rather than pursuing a single optimal solution, we focuses on identifying multiple satisficing solutions that meet acceptable performance thresholds, ensuring robust policy outcomes in dynamic and uncertain environments.

4.4.2.2 Balancing Abstraction and Action in Policy Designing

As discussed in Howlett and co-authors [6], the hierarchical levels of policy designing categorize policy goals and policy means into distinct levels of abstraction, providing a structured approach to connect strategic visions with actionable measures. At the high level, broad goals such as enhancing agricultural productivity or achieving sustainable practices are defined along with general implementation preferences like precision farming or renewable energy adoption. At the program level, meso-level targets bridge abstract aims and practical actions, such as increasing IoT adoption or reducing water usage, supported by tools like financial incentives, training programs, and regulatory standards. At the ground level, specific targets and calibrations such as deploying IoT hubs or setting subsidy criteria translate these visions into actionable, measurable policy outcomes. In this chapter, we translate the different layers of abstraction into foundational, operational, and social engagement design decisions. This perspective not only informs systems design but also offers systems designers a comprehensive framework grounded in design theory, thereby integrating systematic approaches to achieve robust outcomes.

Foundational, Operational, and Engagement Policy Designing Decisions

In this chapter, we consider the levels of policy designing through the lens of foundational decisions, operational decisions, and engagement and social impact decisions, aligning them with the categories proposed in [6]. This ideation is utilized in the development of Integrated Policy Designing Decision Model (IPDDM) which is explained in Section 4.4.3. By utilizing this perspective, the interconnected nature of policy formulation, implementation, and societal engagement is emphasized. In description that follows; we draw connections between the proposed categories and that used in [6].

Foundational Decisions (High-Level Abstraction)

At the foundational level, the focus is on setting the long-term vision and overarching policy objectives, akin to the "general abstract policy aims" and "implementation preferences" referred in [6]. These decisions form the backbone of policy design, ensuring that macro-level goals guide all subsequent actions.

Example in Agriculture 5.0: A foundational decision might set a national goal to "increase technology adoption in 70% of farms by 2030," with policy means involving the promotion of IoT technologies, initial subsidies, and partnerships with technology providers.

Significance: Establishing clear guiding principles here ensures the policy framework remains adaptable without losing sight of its core objectives.

Operational Decisions (Policy Level/Program Level Operationalization)

Operational decisions translate foundational goals into concrete objectives and tools—the "operationalizable policy objectives" and "tools" as referred in [6]. Detailed planning and selection of instruments occur at this stage.

Example in Agriculture 5.0: An operational decision may set a target to "deploy 1,000 IoT hubs in rural regions within five years," supported by financial incentives, farmer training programs, and resource-efficiency regulations.

Significance: These decisions bridge high-level aspirations with actionable measures, ensuring goals are attainable even amid uncertainties like farmer readiness or technological constraints.

Engagement and Social Impact Decisions (Specific On-the-Ground Measures)

At this level, policies are implemented in ways that directly impact communities, reflecting the "specific policy targets" and "tool calibrations" [6].

Example in Agriculture 5.0: If the target is to "train 10,000 farmers in precision farming by 2025," engagement decisions would include localized awareness campaigns, on-site workshops, and tailored subsidies to make IoT devices affordable.

Significance: Engagement decisions ensure policies are socially inclusive and resilient, with flexibility to address challenges such as stakeholder resistance, limited infrastructure, or economic disruptions.

Tying It Together: Robustness in Multi-Level Policy Designing

Viewing policy designing through these levels—foundational, operational, and engagement—underscores the importance of interconnected and robustness at every stage. Through foundational decisions a robust framework that can adapt to changing high-level conditions is provided, such as shifts in societal values or global trends. Operational decisions require tools that are effective across a range of scenarios and uncertainties. In engagement decisions priority is flexibility and inclusivity, ensuring that on-the-ground measures are not only effective but also socially acceptable and equitable. By considering policy designing as an interconnected process across these levels, policymakers can design robust, well informed policies that remain effective and relevant in the events of uncertainties and variability. This method reinforces the importance of

integrating cyber, physical, and social dimensions into the policymaking process, ensuring that policies are both comprehensive and actionable.

To design policies that are robust and take uncertainty into account, there is a need to develop methods that address the nuances of policymaking. Designing Policy is inherently complex, requiring decision-makers to balance competing objectives, address uncertainties, and manage the interconnected nature of decisions. Traditional approaches have limitations to capture the nuances of these interactions, particularly in multi-stakeholder environments with diverse and evolving goals. Recognizing these challenges, the Coupled Decision Method is proposed as a structured method to navigate the intricacies of policy designing. Policymakers use this method to model and analyze interdependent, dynamic decisions and to resolve conflicts among them.

Through this framework designers represent policy decisions in the classification model we propose, the Integrated Policy Designing Decision Model (IPDDM). This is inspired from the work done by [89] which is related to robust design for mechanical components like gear and shaft. Through the IPDDM designers organize decisions along three axes: decision types (Y), interaction strength (X), and level of abstraction (Z). Together, these dimensions provide a holistic view of the policy landscape, enabling decision-makers to align strategic goals with actionable policies.

4.4.3 The Coupled Decision Method

At the core of the coupled decision method is the Integrated Policy Designing Decision Model (IPDDM), which serves as a blueprint for structuring and analyzing decisions. Please note that IPDDM is an integral part of the coupled decision method. Through this classification model we categorize decisions into distinct types, captures the interactions between them, and organizes them to reflect their roles in the policy process.

Decision Types (Y-axis)

The Y-axis in IPDDM (Figure 4.2) classifies decisions into three types:

Selection Decisions involve choosing discrete options that set the strategic context for the policy framework. These choices define the structure within which continuous variables are optimized.

Example in Agriculture 5.0: Selecting the type of subsidy mechanism—fixed subsidies for all farmers versus sliding scale subsidies based on farm size or readiness. A decision to prioritize IoT

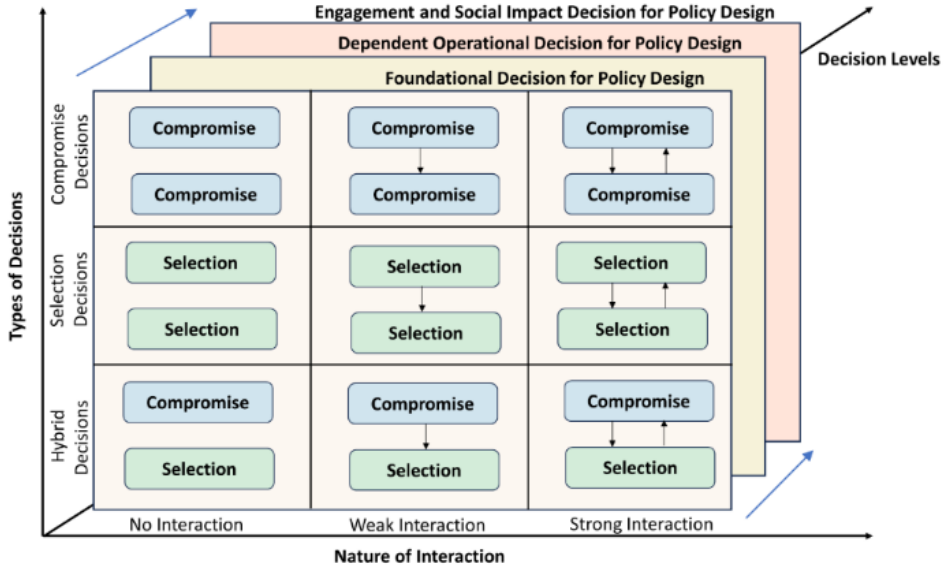


Figure 4.2: Integrated Policy Designing Decision Model (IPDDM)

deployment over autonomous machinery reflects a strategic focus that influences foundational and operational layers.

Significance: Selection decisions ensure that policies align with high-level objectives and stakeholder priorities, embedding strategic intent into the framework.

Compromise Decisions determine the satisficing values of continuous variables, balancing competing objectives.

Example in Agriculture 5.0: Setting the percentage of subsidies for setup and operational costs or determining the scale of IoT deployment. These decisions aim to maximize adoption rates while minimizing costs.

Significance: Compromise decisions operationalize policies by finding trade-offs that satisfy multiple objectives, such as

economic efficiency, social equity, and environmental sustainability.

Hybrid Decisions combine elements of both selection and compromise decisions, allowing for flexibility in addressing complex scenarios.

Example in Agriculture 5.0: A hybrid decision could involve selecting a subsidy mechanism (selection decision) and simultaneously optimizing the percentage allocation (compromise decision). For instance, choosing a sliding scale subsidy and then determining the exact scale for small, medium, and large farms.

Significance: Hybrid decisions bridge strategic intent and operational execution, ensuring that high-level choices are adaptable to specific contexts.

X-Axis: Strength of Interaction

The X-axis in IPDDM (Figure 4.2) represents the strength and directionality of interactions between decisions:

- **No Interaction**: Decisions that operate independently and have no mutual influence.
- **Weak (One-Way) Interaction**: Decisions where information flows in a unidirectional manner. For example, the scale of IoT deployment might depend on subsidy levels, but not vice versa.
- **Strong (Two-Way) Interaction**: Decisions that influence each other bidirectionally, requiring iterative resolution to achieve consistency. For instance, the relationship between training programs and campaign expenditures might be strongly coupled as one reinforces the other.

Designers must record these interactions if they are to trace how a decision at one level affects outcomes at another.

Z-Axis: Decisions at Different Levels of Abstraction

The Z-axis organizes decisions as per different levels of abstraction:

- **Foundational Decisions**: These decisions form the strategic base of the policy, allocating resources and setting priorities. Examples include determining subsidy levels and infrastructure investments.
- **Operational Decisions**: These decisions focus on implementation, ensuring that foundational strategies translate into practical outcomes. Deploying shared machinery centers and organizing training programs fall into this category.
- **Engagement Decisions**: These address stakeholder involvement and long-term sustainability, such as awareness campaigns to drive social acceptance.

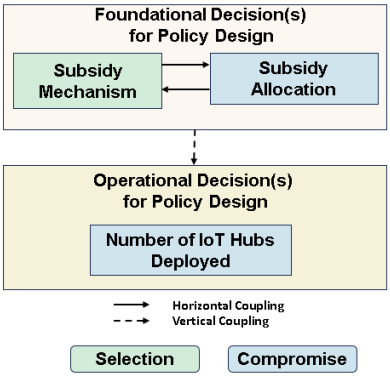


Figure 4.3: Coupled Representation and Modeling of Interactions and Interdependencies Among Decisions Across Varying Abstraction Levels Between Three Decisions

This perspective for looking at different decisions ensures that strategic goals at higher levels of abstraction effectively into actionable plans at lower levels. In Figure 4.3, the interactions between

foundational and operational decisions in the context of IoT deployment and subsidy mechanisms are shown. At the foundational level, a selection decision (subsidy mechanism) and a compromise decision (subsidy allocation) are horizontally coupled, demonstrating a strong interaction where the choice of subsidy mechanism (e.g., fixed, sliding scale, or performance-based subsidies) directly influences the subsidy allocation strategy. Vertically, these foundational decisions impact the operational level compromise decision (Number of IoT Hubs Deployed). Through the choice of subsidy mechanism, designers determine the funding model and overall affordability of IoT deployment, while the subsidy allocation influences the scale and distribution of IoT hubs, ensuring their alignment with policy goals. Through this interconnected structure we can observe how decisions within and across levels dynamically influence each other, ensuring cohesive and aligned policy implementation.

Followed by representing the design decisions and critical aspects of designing policy through IPDDM, we deploy a Compromise Decision Support Problem (cDSP) construct to mathematically model the trade-offs and interdependencies among competing goals. Through the cDSP construct policy designers are able to facilitate the exploration of satisficing design space by balancing conflicting criteria, ensuring that the policy outcomes are robust and align with overarching goals. Through this construct designers can carry systematic analysis of decision variables, enabling policymakers to make informed choices while addressing uncertainties and variability in the decision-making process.

4.5 Uncertainty Management and Incorporating Robustness in Policy Designing

In this chapter we have incorporated robustness in designing policies through a key metric - the Design Capability Index (DCI), as policy goals. Originally developed to measure process capability in engineering, these metrics can be reimagined for evaluating the robustness of policy frameworks. Through DCI, designers assess how well a policy can accommodate variations and uncertainties in its design and implementation. This is the first attempt to utilize this metrics in policy designing and their needs to be additional exploration to understand the utility and the

appropriate use of DCI in designing policies. Just as DCI has been used to measure performance variations in engineering caused by manufacturing fluctuations, it has the potential to be applied to policy designing to evaluate variations in outcomes due to unpredictable factors such as social behaviors, economic shifts, or environmental changes. This needs to be further investigated. Unlike traditional statistical metrics that rely on fixed distributions, the DCI can address scenarios where the deviation in policy performance cannot be easily quantified by statistical methods, making it highly relevant for complex, dynamic policy environments. This adaptation ensures that policy decisions are not only resilient to variability but also capable of maintaining their intended functionality under a range of uncertain conditions.

4.5.1 Design Capability Index

In the context of policy designing, Design Capability Indices (DCIs) can be conceptualized as the safety margin against policy failure due to uncertainties in decision variables. DCIs are an indicator of the robustness of policy goals by measuring their capability to:

1. Meet Policy Objectives: Ensuring that the policy satisfies its intended goals and requirements, even under varying conditions.
2. Accommodate Uncertainty: Evaluating how well the policy can tolerate uncertainties in critical variables, such as fluctuating economic conditions, behavioral responses, or environmental shifts.

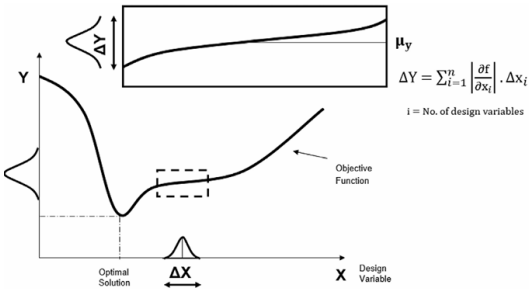


Figure 4.4: Formulation of Uncertainty Bounds Due to Variations In a Design Variable [76]

For instance, if a policy decision involves introducing a subsidy for technology adoption, variations in public uptake or external economic conditions may affect the outcome. DCIs can enable the policy designers to incorporate the uncertainty in the form of variability and model policy goals as DCI goals to attain robust design solutions and policy options.

In Figure 4.4, the mean response (μy) represents the expected policy outcome under normal conditions. At a given point x, variations $\pm\Delta x$ in a policy variable (e.g., subsidy rates) produce an expected deviation (ΔY) in the outcome (e.g., Technology adoption rates). This helps calculate the

maximum expected deviation in policy performance for any given variation in the decision variable.

In Figure 4.5 the mathematical formulations for implementing DCIs as a goal in robust policy designing is outlined. The principles are as follows:

- Smaller is Better: Minimize the target function, such as reducing policy inefficiencies or costs.
- Larger is Better: Maximize desired outcomes, such as increasing policy impact or adoption rates.
- Nominal is Better: Achieve outcomes close to the targeted value, avoiding underachievement (policy failures) and overachievement (resource overutilization).

By incorporating DCIs into policy design, policymakers can develop frameworks that not only meet their goals but also exhibit robustness to uncertainties, ensuring that policies perform reliably across a range of conditions. Through this approach policy designers can have a structured way to measure and incorporate policy robustness, balancing various policy metrics.

Steps for formulating goals as DCI

To formulate goals as Design Capability Indices (DCIs), the process involves two essential steps.

(i) Estimate Response Variation

Using a first-order Taylor series expansion, calculate the response variation (ΔY) due to small changes in the design variables (x). The variation is expressed as:

$$\Delta Y = \sum_{i=1}^{n} \left| \frac{\partial f}{\partial x_i} \right| \Delta x_i$$

Here, *f* represents the performance function, x_i are the design variables, and Δx_i is the variation in each design variable. In this step we capture how changes in variables propagate to influence outcomes. We attempt to quantify how small changes in policy parameters impact overall performance, providing insights into the system's sensitivity to variations check.

(ii) Compute the DCI

Using the mean response *(μy)* from the model *($f_0(x)$)* and the calculated response variation (ΔY),

the DCI is determined. For example, in a "Larger is Better" case (e.g., maximizing adoption rates):

$$DCI = \frac{\mu_y - LRL}{\Delta Y}$$

Here, LRL is the lower requirement limit for the response. A DCI value below 1 indicates that the performance falls outside the acceptable range, while a DCI greater than or equal to 1 ensures that the design is robust to variations within the acceptable limits.

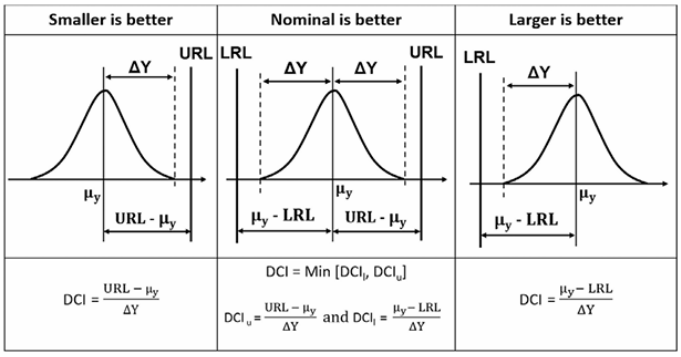


Figure 4.5: Mathematical Constructs for Design Capability Indexes (DCIS) [76]

The Design Capability Index (DCI) is a mathematical metric that quantifies the safety margin against system failure due to uncertainties in design variables. In the context of policy designing, the DCI indicates how robust a policy framework is when faced with variations in critical parameters. A DCI < 1 signifies that the policy's expected performance falls outside the acceptable range of requirements, indicating potential failure under uncertain conditions. Conversely, a DCI ≥ 1 indicates that the policy's specifications meet the performance requirements, demonstrating resilience to uncertainty.

Policy designers should aim to achieve a DCI more than unity as possible, ensuring that a significant portion of the performance deviation remains within the acceptable range of outcomes. Higher DCI values indicate a greater safety margin and a stronger measure of robustness against uncertainty. By targeting a DCI ≥ 1, policymakers can design policies that maintain functionality and effectiveness even when subject to unpredictable factors that introduce variability.

4.6 Demonstration of the Efficacy of The Proposed Framework

In this section we discuss on demonstration of the proposed method, mathematical formulation, and the results obtained by exercising the proposed IPDD, Coupled Decision Method, and the overarching framework on Agriculture 5.0 Test Problem as described in Section 4.3. We use a

synthetic test problem where all the values are *selected to preserve logical consistency.* The details of variables are described in Table 4.2. The goals through the policy are -

Goal 1: Maximize technology adoption rates.
Goal 2: Minimize costs for farmers. = Maximize the covered cost
Goal 3: Improve agricultural productivity.
Goal 4: Reduce environmental impacts.
Goal 5: Enhance social acceptance of new technologies.

For Goal 1, achieving maximum adoption rate, setting up subsidies (X1) are most effective at reducing entry costs while operational subsidies (X2) and we ensure long-term adoption with a larger coefficient than X1. IoT hubs (X3) effected on adoption by aiding decision-making, though the effect reduces logarithmically, and training (X5) helps adoption but ranks second to monetary incentives. Adoption rate is scaled by a denominator of 140 to keep values within a workable range.
For Goal 3, enhancing agricultural productivity, IoT nodes (X3) improve productivity by a small extent, while common equipment (X4) has a more direct and larger effect. Scaling productivity by a factor of 1.3 maintains the productivity index within a viable range.
For Goal 4, minimizing environmental footprint, shared equipment (X4) has a substantially lower environmental footprint with efficient utilization of resources, while IoT nodes (X3) improve utilization of inputs but with lower impact. The environmental footprint is normalized with a denominator of 150.
For Goal 5, maximum social acceptance, the training effect (X5) falls off logarithmically with increasing numbers of people being trained, and awareness campaigns (X6) experience high initial returns. A denominator of 10 will standardize the social acceptance score to a similar scale to other goals.

Table 4.2: System Variables and Its Significance for Agriculture 5.0

X_1 (FD)	Subsidy Percentage - Focus is on the initial financial support for technology adoption through subsidies.	These decisions create the strategic backbone of the policy by

X_2 (FD)	Operational Subsidy Allocation - Ensures the sustainability of technologies by providing operational subsidies.	allocating resources and ensuring long-term feasibility (**FD)**.
X_3 (OD)	Number of IoT Hubs - Deployed Decides the scale and deployment of IoT hubs to establish a technological infrastructure.	These decisions translate foundational strategies into actionable infrastructure and resource access measures .
X_4 (OD)	Number of shared Machinery Centers - Focus is on deploying shared machinery centers to enable resource-sharing among farmers.	
X_5 (E&SI)	Number of Training Sessions - Implements training programs to educate farmers on IoT systems and shared machinery use	these decisions ensure that policies effectively reach stakeholders, fostering social inclusion and practical implementation.
X_6 (E&SI)	Expenditure on campaigns - Funds awareness campaigns to drive adoption and build social acceptance for the technologies.	
FD: Foundational Design Decision; OD: Operational Design Decision; E&SI: Engagement and Social Impact Design Decision		

4.6.1 Mathematical Formulation

Given

- Target values to achieve Goals
- ✓ Target for Goal 1 = 0.75
- ✓ Target for Goal 2 = 2380
- ✓ Target for Goal 3 = 1.9
- ✓ Target for Goal 4 = 1.0
- ✓ Target for Goal 5 = 1.0
- ✓ DCI_G1_Target = 40 (Calculated)
- ✓ DCI_G2_Target = 4 (Calculated)

- Variability in X1 [+- 0.15] (uncertainty)

- Variability in X2 [+- 0.05] (uncertainty)

The uncertainties reflect fluctuations in initial subsidy support due to policy shifts or budget constraints (for X1) and variations in operational subsidies based on evolving financial and operational conditions (for X2).

- Compromise System Parameters

✓ $C_{setup} = 2000$

✓ $C_{oper} = 1000$

The estimated cost of providing setup subsidies per unit and operational subsidies per unit with a higher coefficient requirement.

- Selection System Parameters

Selection of Subsidy Mechanisms

Design Variables	Equity Level (Percentage)	Data Requirement (Scale)	Implementation Complexity (Level
Y1 (Fixed Subsidies)	100% equal for all	Low	Low
Y2 (Sliding-Scale Subsidies)	Varies by 20-50%	Medium	Medium
Y3 (Performance-Based Subsidies)	Depends on 30-80% performance metrics	High	High

Find

- Selection System Variables

  Subsidy Mechanism, Y1, Y2, Y3

- Compromise System Variables X1, X2, X3, X4, X5, X6
- Deviation Variables

Deviation Variable	Description
d_1^+	Overachievement of the adoption rate goal

$\boldsymbol{d_1^-}$	Underachievement of the adoption rate goal
$\boldsymbol{d_2^+}$	Overachievement of the cost reduction goal
$\boldsymbol{d_2^-}$	Underachievement of the cost reduction goal
$\boldsymbol{d_3^+}$	Overachievement of the productivity improvement goal
$\boldsymbol{d_3^-}$	Underachievement of the productivity improvement goal
$\boldsymbol{d_4^+}$	Overachievement of the environmental impact reduction goal
$\boldsymbol{d_4^-}$	Underachievement of the environmental impact reduction goal
$\boldsymbol{d_5^+}$	Overachievement of the social acceptance goal
$\boldsymbol{d_5^-}$	Underachievement of the social acceptance goal
$\boldsymbol{e_1^-}$	Underachievement of MF Goal (Selection Variable)

Satisfy System Constraints

- Selection Design Constraints

Selection Constraint for Subsidy Selection Mechanism

$$\sum_{i=1}^{3} Y_i = 1$$

- Compromise Design Constraints

Budget Constraint

$$1000 * \ X_1 + 2000 \ * \ X_2 + X_3 + 50 * X_4 + 45^* X_5 + \ X_6/10^3 < 9000$$

Robust Solution Constraint on DCI_{G1} Goal

$$DCI_{G1} \geq 1$$

Robust Solution Constraint on DCI_{G2} Goal

$$DCI_{G2} \geq 1$$

Constraints on Deviation Variables

$$d_i^+ \geq 0, d_i^- \geq 0, d_i^+ . d_i^- = 0 \qquad for\ i = 1,2,3,4,5$$

System Goals

- **Goal 1: Maximize Adoption Rate**

$$\frac{(10.X_1 + 15 * X_2 + 2.8 . (\log(1 + X_3) + 0.8.X_5)}{140}$$

$$DCI \geq \frac{\mu_{y-LRL}}{\Delta Y}$$

$$\Delta Y = \left(\frac{10}{140}\right) . (0.15) + \left(\frac{15}{140}\right) . 0.05 = 0.016$$

$$LRL = 0.047857$$

$$DCI_{G1} = \frac{[\frac{(10.X_1 + 15 * X_2 + 2.8 . (\log(1 + X_3) + 0.8.X_5)}{140}] - 0.047857}{0.016}$$

$$DCI\_G1_{Target} = 40$$

- **Goal 2: Minimize Farmer Cost**

$$(1 - X_1) . C_{setup} + (1 - X_2). C_{oper}$$

For programming convenience, we convert this equation to Maximization form –

Goal 2: Maximize Covered Cost by farmers

$$(X_1) . C_{setup} + (X_2). C_{oper}$$

$$DCI \geq \frac{\mu_{y-LRL}}{\Delta Y}$$

$$\Delta Y = 2000 . 0.15 + 1000 . 0.05 = 350$$

$$LRL = 980$$

$$DCI_{G2} = \frac{[(X_1 . 2000 + X_2 . 1000) - 980]}{350}$$

$$\mathrm{DCI\_G2_{Target}} = 4$$

- **Goal 3: Maximize Agricultural Productivity**

$$1.3.\,(\,1 + 0.0002\,.X_3 + 0.003\,.X_4\,)$$

$$0.00026.X_3 + 0.0039\,.X_4$$

- **Goal 4: Maximize Positive Env. Impact (Minimize Environmental Impact)**

$$\log\left[\frac{\mathrm{X_3} + 10.\mathrm{X_4}}{150}\right]$$

- **Goal 5: Maximize Social Acceptance**

$$\left[\frac{\log(1+\mathrm{X_5}).\log(1+\ \mathrm{X_6})}{10}\right]$$

- **Goal 6: Maximize Merit Function**

$$\mathrm{MF_i}(\mathrm{X_1},\mathrm{X_3},\mathrm{X_6})\ \ \mathrm{Y_i} + \ \mathrm{e_1^-} - \ \mathrm{e_1^+} = 1$$

Bounds

$$0.4 < \mathrm{X1} < 0.85$$

$$0.18 < \mathrm{X2} < 0.45$$

$$0 < \mathrm{X3} < 1000$$

$$0 < \mathrm{X4} < 50$$

$$0 < \mathrm{X5} < 100$$

$$0 < \mathrm{X6} < 100{,}000$$

Objective Function

$$Z = [e_1^- + \ \sum_{i=1}^{5} w_i(\,\mathrm{d_i^+} + \ \mathrm{d_i^-})]\ ;\ \sum_{i=1}^{5} w_i = 1$$

(**Note** – *The selection decision is included to demonstrate mathematically. The analysis has not been included.)*

4.6.2 Demonstration of Results

In this chapter, we have outlined key tools and methods commonly employed in the field of policy designing. Building on these approaches, by using cDSP and DSIDES designers can integrate multiple scenarios that align closely with the judgment and expertise of policy designers. These scenarios are designed to reflect plausible situations that policymakers frequently encounter, enabling them to manage the complexities of decision-making in a way that is both practical and contextually relevant to policy designing. This integration not only enhances the robustness of policy frameworks but also ensures that they are better suited to address the dynamic challenges

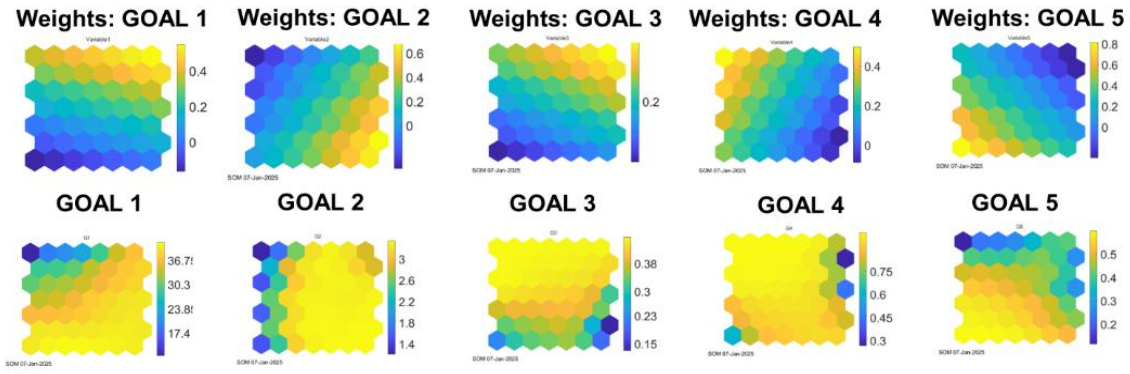


Figure 4.6: Input-Output Component Planes for Test Problem

inherent in the policymaking process.

Table 4.3: Sample Scenarios for Weights

Scenarios	W1	W2	W3	W4	W5
1	0.2	0.2	0.2	0.2	0.2
12	0	0	0	0	1
16	1	0	0	0	0
25	0.90	0.10	0	0	0
33	0.43	0.09	0.03	0.30	0.15
61	0.01	0.02	0.26	0.17	0.54

We have examined 61 different scenarios within the context of policy designing using cDSP. These scenarios are carefully selected to ensure that the policy designing space was

adequately sampled for thorough exploration through iSOM. The sample scenarios are as shown in Table 4.3.

The scenarios represent different configurations of policy priorities, reflecting how policy designers allocate weights based on the preference of each goal. In Scenario 1, policy designers assign equal weight to all goals, indicating a balanced approach where all objectives are considered equally important. This approach may be applied in contexts where a comprehensive or holistic solution is needed, without prioritizing one goal over another. In Scenario 12, policy designers allocate all weight to a single goal (W5), signaling that one objective is of critical importance. This approach might be chosen in cases were addressing a specific issue, such as a crisis or a highly sensitive policy area, requires exclusive attention. In Scenario 16, the designers place all weight on W1, showing a singular focus on one priority while disregarding the others. This configuration suggests a highly targeted approach, driven by the need to achieve a particular policy outcome. In Scenario 25, policy designers heavily prioritize W1 with a smaller allocation to W2, creating a weighted approach that emphasizes one primary goal while giving minimal consideration to a secondary objective. This scenario might reflect a situation where a dominant policy aim is supported by an auxiliary or complementary goal. Finally, in Scenario 33,

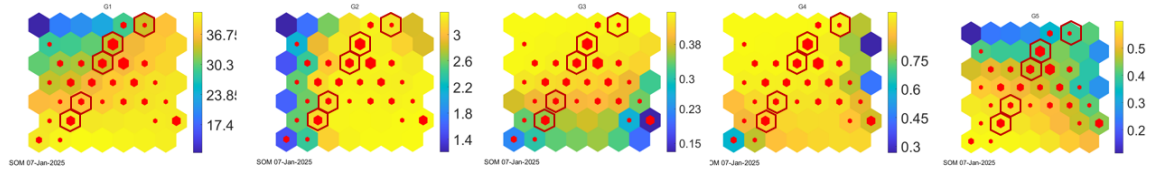


Figure 4.7: Region of Interest Shown by the Red Border Hexagons and Hits Shown By Red Dots

designers distribute weights across all goals, with W1 and W4 receiving the most emphasis but still allowing for contributions from other objectives. Through these scenarios we observe how policy designers strategically allocate weights to goals based on the context and situational demands, enabling tailored and robust policy designs that align with specific objectives. The 61 scenarios (as outlined in Table 4.3) are utilized in the formulation of the deviation function within the cDSP framework. We exercise cDSP construct for all the scenarios using DSIDES. Based on the input wight values and output goal values we run iSOM algorithm to visualize the design space. In Figure 4.6 the component planes for all the goals, providing a visual representation of the weights assigned to each goal is observed. The input is the weight scenarios for each goal.

4.6.3 Interpretation and Discussion on Results

Each hexagonal grid represents clusters of nodes (units) with specific weight values mapped for the given goals. The color gradients indicate the magnitude of weights, where yellow indicates higher values and blue indicates lower values for each goal. This helps identify regions in the iSOM map that correspond to different magnitudes of influence for the respective goals.

As seen from the output component planes, it is evident that the high values for Technology Adoption align with the higher values of Agricultural Productivity. We can also see not very high values for Goal 3 which is maximize agricultural productivity. G1 – X1, X2, X3, X5 and Goal 3 is a function of X4. We formulated this problem to demonstrate that policymakers do not necessarily

Table 4.4: Solution Points Selected in RoI as Per Hits (DCI Converted Values For G1 And G2)

Scn = Scenarios

Scn	W1	W2	W3	W4	W5	G1	G2	G3	G4	G5	X1	X2	X3	X4	X5	X6
2	0.10	0.10	0.1	0.1	0.6	0.68	2145.5	0.3	0.9	0.58	0.84	0.449	997.3	23.3	90.75	996.09
39	0.11	0.02	0.23	0.01	0.63	0.73	2145.5	0.3	0.8	0.60	0.84	0.449	997.3	15.1	99.73	998.69
62	0.01	0.02	0.26	0.17	0.54	0.73	2145.5	0.3	0.8	0.60	0.84	0.449	997.39	15.1	99.73	998.69
6	0.60	0.10	0.1	0.1	0.1	0.60	2149	0.4	1	0.46	0.85	0.45	1000	50	76.98	285.71
31	0.45	0.06	0.21	0.26	0.01	0.63	2149	0.4	1	0.35	0.85	0.45	1000	50	81.850	66.716
30	0.38	0.06	0.14	0.18	0.24	0.56	2149	0.4	1	0.520312	0.85	0.45	1000	50	68.518	666.667
48	0.06	0.03	0.46	0.28	0.17	0.56	2149	0.4	1	0.520312	0.85	0.45	1000	50	68.5185	666.667

maximize agricultural productivity when they maximize technology adoption rates because farmers share resources under locally specific conditions. This is a critical social factor for the policy to be effective. Clusters with high Goal 3 values and moderate Goal 2 values suggest that higher productivity might justify slightly higher costs for farmers, especially if it leads to long-term benefits.

Based on the component planes we identify the Regions of Interest (RoI) as shown in Figure 4.7. This depends on the judgement of the policy designers and the goals to be achieved through the policy design. The solution points are determined by identifying the sample points that fall within the Region of Interest (RoI) to evaluate how well the goals are achieved in the context of designing policies. In this framework, the solution points represent potential configurations or strategies that align with the objectives through policy designing. Red dots indicate samples mapped to specific

nodes, each representing a policy scenario. When a node is selected as the Best Matching Unit (BMU), it is considered to have a "hit." The size of the red dots reflects the number of hits, with larger dots indicating that more samples align with that particular policy configuration.

Within the RoI, nodes with hits correspond to six solution points. These solution points provide valuable insights into policy designing strategies and their alignment with the desired goals. The results for each of these solution points are summarized in Table 4.4. In Table 4.4 the Goals 1 and 2 are converted goals.

4.6.4 Sample Policy Design

Let us consider Scenario 39 in the pursuit of designing robust policies for the Agriculture 5.0 test problem. Before analyzing the specific values, it is essential to understand how robustness is integrated into the policy design using this scenario. For example, in Scenario 39, the technology adoption rate (G1) is 73%. This means that even with variability of ±15% in X1 (subsidy percentage), which can reflect fluctuations in initial subsidy support due to policy shifts or budget constraints, and ±5% in X2 (operational subsidy allocation), which can represent variations in operational subsidies based on evolving financial and operational conditions defined by the policy designer's judgment, the policy will maintain a satisficing solution space for Goal 1 (maximize technology adoption) and Goal 2 (minimize costs for farmers). This robustness is ensured by using the Design Capability Index (DCI) for Goals 1 and 2, where we set the minimum requirements as the lower requirement limit. By defining the DCI-Target values and the DCI constraint greater than 1, we ensure that even with the mentioned uncertainty, we are able to account for uncertainties and variability in key policy variables.

Additionally, it is important to note that the values for G1 and G2 represent the worst-case outcomes. This is due to the conservative calculation approach employed in the cDSP and DCI formulations. As a result, the policies designed under this framework are inherently robust, ensuring effectiveness even under varying conditions and uncertainties.

Here is a sample policy design (Table 4.5) through the method proposed and demonstrated in this chapter.

Table 4.5: Sample Policy Design	
Policy Goals	
Goals	**Expected Outcome**
Maximize technology adoption rates (G1)	73%.
Minimize costs for farmers (G2): Equivalent to maximizing the covered cost,	set at $2145.5.
Improve agricultural productivity (G3):	30%.
Reduce environmental impacts (G4)	Achieve an 80% overall positive environmental impact.
Enhance social acceptance of new technologies (G5)	Achieve a 60% acceptance rate.
Maximize technology adoption rates (G1)	73%.
Key Policy Design Variables	
X1 (Subsidy Percentage)	84% of costs are covered as subsidies.
X2 (Operational Subsidy Allocation)	49% of the budget is allocated to operational subsidies.
X3 (Number of IoT Hubs Deployed)	997 hubs.
X4 (Number of Shared Machinery Centers)	15 centers.
X5 (Number of Training Sessions)	99 sessions.
X6 (Expenditure on Campaigns)	$998,690.

Policy Design Framework

In the following part we have demonstrated how can an actual policy be designed by the insights gathered through the proposed method (and the framework).

Foundational Decisions (Strategic Goals)

Subsidy Allocation (X1): Allocate 84% of costs as subsidies to maximize technology adoption (G1), with ±15% variability accommodated.

Operational Subsidy Allocation (X2): Allocate 49% of the total budget to operational subsidies, directly supporting G2 while allowing ±5% flexibility.

Operational Decisions (Infrastructure and Productivity)

IoT Hubs Deployment (X3): Deploy 997 IoT hubs across agricultural regions to enhance digital connectivity, improving productivity (G3) and reducing environmental impact (G4).

Shared Machinery Centers (X4): Establish 15 centers to promote resource efficiency and cost-sharing among farmers, supporting G3.

Engagement and Social Impact Decisions

Training Programs (X5): Conduct 99 sessions to educate stakeholders on Agriculture 5.0, enhancing technology adoption (G1) and social acceptance (G5).

Awareness Campaigns (X6): Invest $998,690 in campaigns to raise public awareness, addressing environmental (G4) and social impacts (G5).

4.7 Closing Remarks for Chapter 4

Our objective in this chapter is to demonstrate the potential of a robust, design-centric method for developing effective policies in uncertain, interdependent environments. We address the question: *"What methods and principles effectively account for uncertainty in designing robust policies for complex systems?"* To answer this, we propose an integrated approach that employs a decision classification scheme (IPDDM), a Coupled Decision Method based on cDSP, and visualization via iSOM to evaluate policy performance across diverse future scenarios.

A central premise is that policy design functions as an evolving cyber-physical-social system, where shifts in social and physical domains ranging from fluctuating stakeholder behaviors to changing resource conditions introduce continuous variability. Rather than rely on static assumptions, we formulate and evaluate policies through five elements:

1. Bridging Interdependent Decisions and Coherent Outcomes: By explicitly modeling interactions among foundational, operational, and engagement decisions, our method clarifies hidden trade-offs in dynamic contexts.

2. Quantifying Robustness in Dynamic Contexts: We demonstrate how the Design Capability Index (DCI) captures a policy's robustness, ensuring solutions remain robust as external conditions shift.
3. Facilitating Transparent, Inclusive Policymaking: By using visualization tools such as interpretable Self-Organizing Maps, diverse stakeholders understand complexities and converge on robust solutions.
4. Embracing Satisficing over Incomplete Optimization: Through our approach we seek acceptable compromises that reflect the multifaceted, value-laden nature of policymaking.
5. Providing a Replicable, Adaptable Framework: Although illustrated through Agriculture 5.0, our method is applicable across domains, from public health to urban planning, thus, moving beyond static, siloed approaches.

We opened this chapter by setting out to address RQ2a and to examine Hypothesis 2a. Within the scope of the Agriculture 5.0 problem we study, the evidence we present is consistent with that hypothesis: classifying decisions through the IPDDM surfaces that three of the six decision variables interact bidirectionally and are better formulated as coupled than as sequential cDSPs, and the Design Capability Index constraints indicate that the satisficing solution space so identified holds across the thirty-three stakeholder priority scenarios we examine. On that basis we offer the new knowledge we set out to produce: a coupled, multi-level decision method anchored in the IPDDM and Design Capability Indices for designing robust satisficing strategies in an E-CPSS under uncertainty. We take this to contribute D(si), the intra-state decision map and the second formal component of the DSIM. With respect to verification and validation, we read the Frame of Reference as offering theoretical structural validity, the Agriculture 5.0 problem as a reasonable vehicle for empirical structural validity given its multi-level and multi-stakeholder structure, and the demonstrated robustness of the satisficing region as empirical performance validity within the bounds of this single problem; theoretical performance validity we defer to Chapter 9.

D(si) characterizes a single state. An E-CPSS does not stand still: the decisions we structure for the present must remain traceable to the outcomes we seek at the next state, and a decision map calibrated to current conditions erodes as the system evolves. What we do not yet address is this temporal dimension. That is the work of Chapter 5, where we move from D(si) to D(si→si+1), the

inter-state decision map, and work backward from desired future states across the system's transitions.

References

[1] Howlett, M., and Mukherjee, I., 2018, "Introduction: The Importance of Policy Design: Effective Processes, Tools and Outcomes," *Routledge Handbook of Policy Design*, Routledge, pp. 3–19.

[2] Schön, D. A., 1988, "Designing: Rules, Types and Words," *Design Studies*, 9(3), pp. 181–190. doi: 10.1016/0142-694X(88)90047-6

[3] Gero, J. S., 1990, "Design Prototypes: A Knowledge Representation Schema for Design," *AI Magazine*, 11(4), pp. 26–36.

[4] Gero, J. S., and Kannengiesser, U., 2004, "The Situated Function–Behaviour Structure Framework," *Design Studies*, 25(4), pp. 373–391.

[5] Hudson, B., Hunter, D., and Peckham, S., 2019, "Policy Failure and the Policy Implementation Gap: Can Policy Support Programs Help?" *Policy Design and Practice*, 2(1), pp. 1–14. doi: 10.1080/25741292.2018.1540378

[6] Howlett, M., 2009, "Governance Modes, Policy Regimes and Operational Plans: A Multi-Level Nested Model of Policy Instrument Choice and Policy Design," *Policy Sciences*, 42, pp. 73–89.

[7] Howlett, M., 2018, "The Contexts and Components of Policy Design: Governance Modes and Policy Regimes," *Routledge Handbook of Policy Design*, Routledge, pp. 20–33.

[8] Daviter, F., 2019, "Policy Analysis in the Face of Complexity: What Kind of Knowledge to Tackle Wicked Problems?" *Public Policy and Administration*, 34(1), pp. 62–83.

[9] Nutley, S. M., Walter, I., and Davies, H. T. O., 2007, *Using Evidence: How Research Can Inform Public Services*, 1st ed., Bristol University Press. doi: 10.2307/j.ctt9qgwt1

[10] Arts, B., and van Tatenhove, J., 2004, "Policy and Power: A Conceptual Framework Between the 'Old' and 'New' Policy Idioms," *Policy Sciences*, 37, pp. 339–356.

[11] Lasswell, H. D., 1958, *Politics: Who Gets What, When, How*, Meridian, New York.

[12] Berger, P. L., and Luckmann, T., 1966, *The Social Construction of Reality: A Treatise in the Sociology of Knowledge*, Doubleday & Company, New York.

[13] Stone, D. A., 1988, *Policy Paradox and Political Reason*, Scott, Foresman, Glenview, IL.

[14] Bhalerao, M. J., Honeycutt, W. T., Das, A. K., Allen, J. K., and Mistree, F., 2023, "Framing Wicked Problems Through Evidentiary and Interpretative Analysis," *Proceedings of the ASME 2023 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 3B: 49th Design Automation Conference (DAC)*, ASME, Boston, MA, USA, Paper No. V03BT03A005. https://doi.org/10.1115/DETC2023-117285

[15] Bhalerao, M. J., 2024, "On Designing Public Policy as an Evolving Cyber Physical Social System," M.S. thesis, University of Oklahoma. https://shareok.org/handle/11244/340292

[16] Haynes, J. P. E., Bhalerao, M. J., Honeycutt, W. T., Allen, J. K., and Mistree, F., 2024, "Predictive Modeling for Public Policy Design: The Impact of Artificial Lights at Night (ALAN) on Bird Strikes," *Proceedings of the ASME 2024 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 2B: 44th Computers and Information in Engineering Conference (CIE)*, ASME, Washington, DC, USA, Paper No. V02BT02A021. https://doi.org/10.1115/DETC2024-143780

[17] Rosenhead, J., 1996, "What's the Problem? An Introduction to Problem Structuring Methods," *Interfaces*, 26(6), pp. 117–131.

[18] Lasswell, H. D., and Lerner, D., 1951, "The Policy Orientation," *The Policy Sciences: Recent Developments in Scope and Method*, Stanford University Press, Stanford, CA, pp. 3–15.

[19] Howlett, M., and Mukherjee, I., 2014, "Policy Design and Non-Design: Towards a Spectrum of Policy Formulation Types," *Politics and Governance*, 2(2), pp. 57–71.

[20] Howlett, M., 2019, "Understanding the Role of Policy Design in Contemporary Policy-Making," *Designing Public Policies: Principles and Instruments*, Chap. 1, Routledge.

[21] Majone, G., 1975, "On the Notion of Political Feasibility," *European Journal of Political Research*, 3, pp. 259–274.

[22] Gilabert, P., and Lawford-Smith, H., 2012, "Political Feasibility: A Conceptual Exploration," *Political Studies*, 60(4), pp. 809–825.

[23] Bailey, J., and Lloyd, P., 2016, "The Introduction of Design to Policymaking: Policy Lab and the UK Government: Design + Research + Society - Future–Focused Thinking," *Proceedings of DRS Design Research Society 50th Anniversary Conference*, Vol. 1, pp. 3619–1635. doi: 10.21606/drs.2016.314

[24] May, P. J., 2003, "Policy Design and Implementation," *Handbook of Public Administration*, B. G. Peters and J. Pierre, eds., Sage, Beverly Hills, CA, pp. 223–233.

[25] Bobrow, D. B., 2006, "Policy Design: Ubiquitous, Necessary and Difficult," *Handbook of Public Policy*, B. G. Peters and J. Pierre, eds., Sage, London, pp. 75–96.

[26] Weaver, R. K., 2009, "Target Compliance: The Final Frontier of Policy Implementation," Brookings Institution, Washington, DC. www.brookings.edu/research/papers/2009/09/30-compliance-weaver

[27] Weaver, R. K., 2010, "But Will It Work? Implementation Analysis to Improve Government Performance," Brookings Institution, Washington, DC. www.brookings.edu/research/papers/2010/02/implementation-analysis-weaver

[28] Montpetit, É., 2003, "Public Consultations in Policy Network Environments," *Canadian Public Policy*, 29(1), pp. 95–110.

[29] Linder, S. H., and Peters, B. G., 1989, "Instruments of Government: Perceptions and Contexts," *Journal of Public Policy*, 9(1), pp. 35–58.

[30] Allen, B., and Ostrom, V., 2014, *The Intellectual Crisis in American Public Administration*, 3rd ed., The University of Alabama Press, Tuscaloosa, AL.

[31] Capano, G., and Woo, J. J., 2018, "Designing Policy Robustness: Outputs and Processes," *Policy and Society*, 37(4), pp. 422–440.

[32] Gándara, D., and Rutherford, A., 2018, "Mitigating Unintended Impacts? The Effects of Premiums for Underserved Populations in Performance-Funding Policies for Higher

Education," *Research in Higher Education*, 59, pp. 681–703. https://doi.org/10.1007/s11162-017-9483-x

[33] Ostrom, E., 2010, "Beyond Markets and States: Polycentric Governance of Complex Economic Systems," *American Economic Review*, 100(3), pp. 641–672. doi: 10.1257/aer.100.3.641

[34] Capano, G., and Lepori, B., 2024, "Designing Policies That Could Work: Understanding the Interaction Between Policy Design Spaces and Organizational Responses in Public Sector," *Policy Sciences*. doi: 10.1007/s11077-024-09521-0

[35] Bodin, Ö., 2017, "Collaborative Environmental Governance: Achieving Collective Action in Social-Ecological Systems," *Science*, 357(6352), p. eaan1114.

[36] Dessai, S., Hulme, M., Lempert, R., and Pielke, R., 2009, "Do We Need Better Predictions to Adapt to a Changing Climate?" *Eos*, 90(13), pp. 111–112.

[37] Kwakkel, J. H., Eker, S., and Pruyt, E., 2016, "How Robust Is a Robust Policy? Comparing Alternative Robustness Metrics for Robust Decision-Making," *Robustness Analysis in Decision Aiding, Optimization, and Analytics*, pp. 221–237.

[38] Lempert, R. J., Popper, S., and Bankes, S., 2003, *Shaping the Next One Hundred Years: New Methods for Quantitative, Long-Term Policy Analysis*, RAND Corporation, Santa Monica, CA.

[39] Walker, W. E., Haasnoot, M., and Kwakkel, J. H., 2013, "Adapt or Perish: A Review of Planning Approaches for Adaptation Under Deep Uncertainty," *Sustainability*, 5(3), pp. 955–979.

[40] Hamarat, C., Kwakkel, J. H., and Pruyt, E., 2013, "Adaptive Robust Design Under Deep Uncertainty," *Technological Forecasting and Social Change*, 80(3), pp. 408–418.

[41] Lempert, R. J., and Collins, M. T., 2007, "Managing the Risk of Uncertain Threshold Responses: Comparison of Robust, Optimum, and Precautionary Approaches," *Risk Analysis*, 27(4), pp. 1009–1026.

[42] Kwakkel, J. H., and Pruyt, E., 2013, "Exploratory Modeling and Analysis, an Approach for Model-Based Foresight Under Deep Uncertainty," *Technological Forecasting and Social Change*, 80(3), pp. 419–431.

[43] Bryant, B. P., and Lempert, R. J., 2010, "Thinking Inside the Box: A Participatory, Computer-Assisted Approach to Scenario Discovery," *Technological Forecasting and Social Change*, 77(1), pp. 34–49.

[44] Bai, X., Tsai, W. T., Paul, R., Feng, K., and Yu, L., 2002, "Scenario-Based Modeling and Its Applications," *Proceedings of the Seventh IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS 2002)*, pp. 253–260.

[45] Mandegari, S., Bhalerao, M. J., Allen, J. K., and Mistree, F., 2024, "Predicting Future States of Evolving Systems," *Proceedings of the ASME 2024 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 3B: 50th Design Automation Conference (DAC)*, ASME, Washington, DC, USA, Paper No. V03BT03A007. https://doi.org/10.1115/DETC2024-146340

[46] Howlett, M., and Ramesh, M., 2023, "Designing for Adaptation: Static and Dynamic Robustness in Policy-Making," *Public Administration*, 101(1), pp. 23–35.

[47] Howlett, M., 2019, "The Temporal Dimension(s) of Policy Designs: Resilience, Robustness and the Sequencing of Tools in Policy Mixes," *International Review of Public Policy*, 1(1), pp. 27–45.

[48] McDaniel, R. R., and Driebe, D. J., 2005, *Uncertainty and Surprise in Complex Systems: Questions on Working the Unexpected*, Springer, Berlin, Germany.

[49] Walker, W. E., Lempert, R., and Kwakkel, J. H., 2013, "Deep Uncertainty," *Encyclopedia of Operations Research and Management Science*, 3rd ed., S. Gass and M. Fu, eds., Springer, Berlin, Germany.

[50] Comfort, L. K., Boin, A., and Demchak, C. C., eds., 2010, *Designing Resilience: Preparing for Extreme Events*, University of Pittsburgh Press, Pittsburgh, PA.

[51] Lempert, R. J., Popper, S., and Bankes, S., 2003, *Shaping the Next One Hundred Years: New Methods for Quantitative, Long-Term Policy Analysis*, Report No. MR-1626-RPC, RAND, Santa Monica, CA, USA.

[52] Dessai, S., and Hulme, M., 2007, "Assessing the Robustness of Adaptation Decisions to Climate Change Uncertainties: A Case Study on Water Resources Management in the East of England," *Global Environmental Change*, 17, pp. 59–72.

[53] Guo, L., Allen, J. K., and Mistree, F., 2024, "Optimize or Satisfice in Engineering Design?" *Research in Engineering Design*, 35, pp. 239–267. https://doi.org/10.1007/s00163-023-00431-5

[54] McInerney, D., Lempert, R., and Keller, K., 2012, "What Are Robust Strategies in the Face of Uncertain Climate Threshold Responses," *Climate Change*, 112, pp. 547–568.

[55] Bobrow, D. B., and Dryzek, J. S., 1987, *Policy Analysis by Design*, University of Pittsburgh Press, Pittsburgh, PA.

[56] Howlett, M., and Rayner, J., 2017, "Patching Versus Packaging in Policy Formulation: Assessing Policy Portfolio Design," *Handbook of Policy Formulation*, M. Howlett and I. Mukherjee, eds., Edward Elgar, Cheltenham, pp. 112–127.

[57] Carlson, J. M., and Doyle, J., 2002, "Complexity and Robustness," *Proceedings of the National Academy of Sciences*, 99(Suppl. 1), pp. 2538–2545.

[58] Jen, E., ed., 2005, *Robust Design: A Repertoire of Biological, Ecological, and Engineering Case Studies*, 1st ed., Oxford University Press, New York, NY.

[59] Anderies, J. M., Folke, C., Walker, B., and Ostrom, E., 2013, "Aligning Key Concepts for Global Change Policy: Robustness, Resilience, and Sustainability," *Ecology and Society*, 18(2), p. 8.

[60] Goodin, R., ed., 1998, *The Theory of Institutional Design*, Cambridge University Press, Cambridge.

[61] Leeson, P., and Subrick, J. R., 2006, "Robust Political Economy," *Review of Austrian Economics*, 19(2–3), pp. 107–111.

[62] Ostrom, E., 1990, *Governing the Commons: The Evolution of Institutions for Collective Action*, Cambridge University Press, Cambridge.

[63] Chermack, T. J., 2004, "Improving Decision-Making With Scenario Planning," *Futures*, 36(3), pp. 295–309.

[64] Mietzner, D., and Reger, G., 2005, "Advantages and Disadvantages of Scenario Approaches for Strategic Foresight," *International Journal of Technology Intelligence and Planning*, 1(2), pp. 220–239.

[65] Li, M., Nemani, V. P., Liu, J., Lee, M. A., Ahmed, N., Kremer, G. E., and Hu, C., 2021, "Reliability-Informed Life Cycle Warranty Cost and Life Cycle Analysis of Newly Manufactured and Remanufactured Units," *Journal of Mechanical Design*, 143(11), p. 112001.

[66] Trebilcock, M., and Hartle, D. G., 1982, "The Choice of Governing Instrument," *International Review of Law and Economics*, 2, pp. 29–46.

[67] Peters, B. G., and Van Nispen, F. K. M., eds., 1998, *Public Policy Instruments: Evaluating the Tools of Public Administration*, Edward Elgar, New York.

[68] Salamon, L. C., 2002, *The Tools of Government: A Guide to the New Governance*, Oxford University Press, New York.

[69] Howlett, M., Kim, J., and Weaver, P., 2006, "Assessing Instrument Mixes Through Program- and Agency-Level Data: Methodological Issues in Contemporary Implementation Research," *Review of Policy Research*, 23(1), pp. 129–151.

[70] Keohane, N. O., Revesz, R. L., and Stavins, R. N., 1998, "The Choice of Regulatory Instruments in Environmental Policy," *The Harvard Environmental Law Review*, 22, pp. 313–367.

[71] Veggeland, N., 2008, "Path Dependence and Public Sector Innovation in Regulatory Regimes," *Scandinavian Political Studies*, 31(3), pp. 268–290.

[72] Lodge, M., 2008, "Regulation, the Regulatory State and European Politics," *West European Politics*, 31(1–2), pp. 280–301.

[73] Kooiman, J., 2008, "Exploring the Concept of Governability," *Journal of Comparative Policy Analysis*, 10(2), pp. 171–190.

[74] Lempert, R. J., Bryant, B. P., and Bankes, S., 2008, "Comparing Algorithms for Scenario Discovery," Report No. WR-557-NSF, RAND, Santa Monica, CA, USA.

[75] Gerst, M. D., Wang, P., and Borsuk, M. E., 2012, "Discovering Plausible Energy and Economic Futures Under Global Change Using Multidimensional Scenario Discovery," *Environmental Modelling & Software*. doi: 10.1016/j.envsoft

[76] Choi, H.-J., Austin, R., Allen, J. K., McDowell, D. L., Mistree, F., and Benson, D. J., 2005, "An Approach for Robust Design of Reactive Powder Metal Mixtures Based on Non-Deterministic Micro-Scale Shock Simulation," *Journal of Computer-Aided Materials Design*, 12(1), pp. 57–58.

[77] Allen, J. K., Seepersad, C., Choi, H., and Mistree, F., 2006, "Robust Design for Multiscale and Multidisciplinary Applications," *Journal of Mechanical Design*, 128(4), pp. 832–843.

[78] Seepersad, C. C., Allen, J. K., McDowell, D. L., and Mistree, F., 2006, "Robust Design of Cellular Materials With Topological and Dimensional Imperfections," *Journal of Mechanical Design*, 128(6), pp. 1285–1297.

[79] Chen, W., Allen, J. K., and Mistree, F., 1993, "Hierarchical Selection in Gas Turbine Maintenance Management," *ASME Advances in Design Automation*, ASME DE, Vol. 65, pp. 87–96.

[80] Karandikar, H., and Mistree, F., 1993, "Modelling Concurrency in the Design of Composite Structures," *Structural Optimization: Status and Promise*, M. P. Kamat, ed., AIAA, pp. 769–806.

[81] Vesanto, J., 1999, "SOM-Based Data Visualization Methods," *Intelligent Data Analysis*, 3(2), pp. 111–126.

[82] Yin, H., 2008, "The Self-Organizing Maps: Background, Theories, Extensions and Applications," *Computational Intelligence: A Compendium*, Springer, pp. 715–762.

[83] Kohonen, T., and Somervuo, P., 1998, "Self-Organizing Maps of Symbol Strings," *Neurocomputing*, 21(1–3), pp. 19–30.

[84] Obayashi, S., and Sasaki, D., 2003, "Visualization and Data Mining of Pareto Solutions Using Self-Organizing Map," *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, pp. 796–809.

[85] Thole, S. P., and Ramu, P., 2020, "Design Space Exploration and Optimization Using Self-Organizing Maps," *Structural and Multidisciplinary Optimization*, 62(3), pp. 1071–1088.

[86] Nagar, D., Ramu, P., and Deb, K., 2023, "Visualization and Analysis of Pareto Optimal Fronts Using Interpretable Self-Organizing Map (ISOM)," *Swarm and Evolutionary Computation*, 76, p. 101202.

[87] Nellippallil, A. B., Rangaraj, V., Gautham, B., Singh, A. K., Allen, J. K., and Mistree, F., 2018, "An Inverse, Decision-Based Design Method for Integrated Design Exploration of Materials, Products, and Manufacturing Processes," *Journal of Mechanical Design*, 140(11), p. 111403.

[88] Baby, M., Guptan, A., Broussard, J., Allen, J. K., Mistree, F., and Nellippallil, A. B., 2024, "A Decision Support Framework for Robust Multilevel Co-Design Exploration of Manufacturing Supply Networks," *Journal of Mechanical Design*, 146(11), p. 111704.

[89] Sharma, G., Allen, J. K., and Mistree, F., 2021, "A Method for Robust Design in a Coupled Decision Environment," *Design Science*, 7, p. e23. doi: 10.1017/dsj.2021.22

[90] Gupta, H., Bhalerao, M. J., Honeycutt, W. T., Allen, J. K., and Mistree, F., 2025, "On Designing Evolving Cyber–Physical–Social Systems: A Decision-Based Design Perspective," *Journal of Computing and Information Science in Engineering*, 25(12), p. 120803.

[91] Amorocho-Daza, G., Gandolfi, C., and Cremades, R., 2025, "A Model-Based Policy Analysis Framework for Social-Ecological Systems," *Ecological Modelling*, 489, p. 110390. https://doi.org/10.1016/j.ecolmodel.2024.110390

[92] Athanasoglou, S., van den Bergh, J. C. J. M., and Filar, J. A., 2021, "A Satisficing Framework for Environmental Policy Under Model Uncertainty," *Environmental Modeling & Assessment*, 26(5), pp. 613–628. https://doi.org/10.1007/s10666-021-09761-x

[93] Liu, J., Shvetsova, O., and Dube, O., 2024, "Modeling and Testing Strategic Interdependence and Tipping in Public Policy Implementation," *Proceedings of the National Academy of Sciences*, 121(42), p. e2414041121. https://doi.org/10.1073/pnas.2414041121

[94] He, W., Huang, Q., Cai, X., Guo, S., Liu, P., and Li, Y., 2024, "An Explicit Robust Optimization Framework for Multipurpose Cascade Reservoir Operation Under Inflow Uncertainty," *Environmental Modelling & Software*, 175, p. 105524. https://doi.org/10.1016/j.envsoft.2024.105524

[95] Motlaghzadeh, K., Yazdanpanah, H., Azadi, H., and Zarafshani, K., 2023, "A Multi-Agent Decision-Making Framework for Evaluating Water and Environmental Resources Management Scenarios Under Climate Change," *Science of the Total Environment*, 875, p. 162963. https://doi.org/10.1016/j.scitotenv.2023.162963

[96] Ganji, M., Bagheri, M., Farhadi, S., and Sabouri, A., 2025, "Comprehensive Study on Hydrogen Production for Sustainable Transportation Planning: Strategic, Techno-Economic, and Environmental Impacts," *Hydrogen*, 6(2), p. 24. https://doi.org/10.3390/hydrogen6020024

[97] Cai, Y., Liu, Y., Wang, H., Zhao, H., and Qiu, Y., 2024, "A Two-Stage Robust Decision-Making Framework (2S-RDM) for Flood Risk Adaptation Under Deep Uncertainty," *Fundamental Research*, 4(1), pp. 84–95. https://doi.org/10.1016/j.fmre.2024.01.006

[98] Deak, S., Dovern, J., and Fendel, R., 2025, "All Models Are Wrong but All Can Be Useful: Robust Policy Design Using Prediction Pools," *Journal of Economic Dynamics and Control*, 156, p. 104799. https://doi.org/10.1016/j.jedc.2024.104799

[99] Fang, X., and Zio, E., 2019, "An Adaptive Robust Framework for the Optimization of the Resilience of Interdependent Infrastructures Under Natural Hazards," *European Journal of Operational Research*, 278(3), pp. 964–979. https://doi.org/10.1016/j.ejor.2019.03.001

[100] Moallemi, E. A., Kwakkel, J. H., and de Haan, F. J., 2018, "Model-Based Multi-Objective Decision Making Under Deep Uncertainty From a Multi-Method Design Lens," *Simulation Modelling Practice and Theory*, 91, pp. 28–49. https://doi.org/10.1016/j.simpat.2018.01.009

[101] Tasic, J. F., Amir, S., and Mahmoud, H., 2019, "Modelling Multilevel Interdependencies for Resilience in Complex Organisation," *Complexity*, 2019, p. 3946356. https://doi.org/10.1155/2019/3946356

[102] Castrejon-Campos, R., Markard, J., Alkemade, F., and Hekkert, M. P., 2020, "Making Policy Mixes More Robust: An Integrative and Interdisciplinary Approach for Clean Energy Transitions," *Energy Research & Social Science*, 60, p. 101316. https://doi.org/10.1016/j.erss.2019.101316

[103] Tasic, J. F., Amir, S., and Mahmoud, H., 2020, "A Multilevel Framework to Enhance Organizational Resilience," *Journal of Risk Research*, 23(8), pp. 1017–1035. https://doi.org/10.1080/13669877.2019.1617340

CHAPTER 5

DESIGNING ADAPTIVE STRATEGIES: GOAL-ORIENTED INVERSE DESIGN ACROSS E-CPSS STATE TRANSITIONS

Background and Context

The United Arab Emirates in 2017 and the UAE in 2030 are not the same system. The labor allocation decisions that were appropriate given the economic structure, energy infrastructure, and governance priorities of 2017 will be irrelevant or counterproductive given the conditions that will prevail by 2030, conditions that are themselves partly the result of what those 2017 decisions set in motion. This is the essential temporal problem of E-CPSS design, and it is one that D(si) by itself cannot solve. A decision map structured for the current state is a good decision map precisely because it is calibrated to the current state. As the system evolves, designers lose confidence in the calibration they established for the earlier state. A designer who can only work forward from current conditions is always reacting to what is already true rather than shaping what will become true.

In Chapter 4, we demonstrate that we can structure decisions within a known state. The question that opens this chapter follows directly: how must those decisions evolve as the system transitions toward a desired future? This is the question that RQ2b addresses. Rather than projecting current conditions forward and asking what outcomes they are likely to produce, we invert the logic: we start from where the system needs to be and derive the decisions that make that future achievable under the uncertainty the system will encounter along the way. This inversion is not merely a mathematical convenience. It is a fundamental reorientation of the design stance, from reactive to prospective, from describing what is to committing to what must become. The Goal-Oriented Inverse Design Method is built around this reorientation.

We develop GOIDM as a six-step framework that formulates a compromise Decision Support Problem at each state in the system's trajectory, applies DCI constraints at every state transition rather than only at the endpoint, and incorporates a feedback mechanism with adaptation rates and

goal updating to keep the designed pathway feasible when observed intermediate performance deviates from projection. The distinction between robustness, which is a property of the design at a given state, and adaptability, which is a property of the process by which the design evolves across states, is made precise and operationalized simultaneously through DCI and through the feedback update rules. We demonstrate the method on the UAE sustainable development test problem, three temporal states spanning 2017 through 2030, four co-evolutionary goals, and eight decision variables with quantified uncertainty bounds, a design timeline long enough and complex enough to expose every limitation of forward-only satisficing.

The output of this chapter is D(si to si+1): the inter-state decision map, which together with D(si) from Chapter 4 completes the Decision Layer of the DSIM. We now have a principled method for reading what the system is doing, structuring decisions within it, and designing pathways through its transitions. Two natural extensions press immediately against the boundary of what these methods can handle. When the number of goals grows beyond four, when the decision space expands to dozens of coupled variables, when hundreds of scenarios make iSOM visualization insufficient for unaided navigation, the satisficing philosophy remains correct but our cognitive capacity as designers cannot keep pace with the solution space. That is the challenge we take up in Chapter 6, where we augment the designer's intelligence with structured generative AI rather than replace it.

Preface

Policymakers face persistent uncertainty, and they often find static approaches insufficient for long-term effectiveness.. In this chapter, we address the following question: How can policymakers design adaptive policies as evolving cyber-physical-social systems (E-CPSS) that remain effective amid dynamic conditions? As our contribution, we develop a Goal-Oriented Inverse Design Method (GOIDM) that enables policymakers to work backward from well-defined long-term objectives to identify feasible, robust policy interventions. Unlike traditional forward optimization or scenario-based approaches, In GOIDM we integrate quantitative robustness metrics directly into the formulation through Design Capability Indices (DCIs), enabling systematic evaluation of policy performance under uncertainty. We extend inverse design principles from engineering into policy contexts, and we define transition states, adaptation rates, and feedback-driven goal updates. We demonstrate GOIDM through a case study on sustainable development in the UAE, where

policymakers balance economic growth, environmental sustainability, and social equity. Using historical data and adaptive feedback, we achieve quantifiable results: an average adaptation rate of 0.042 across policy goals, DCI values exceeding 1.5 for all robustness constraints, and dynamic target adjustments, including a 23.8 percent increase in electricity consumption targets and a 5.3 percent increase in employment goals based on observed 2023 transition state performance. We illustrate how GOIDM enables systematic adaptation, satisficing solution space identification via iSOM, and iterative refinement toward 2030 targets. This framework is generalizable to domains such as resource allocation, supply chain optimization, and public health planning, where adaptive, data-driven decision-making under uncertainty is critical.
Keywords: Cyber Physical Social System, Goal Oriented Design, Adaptive Policy Designing, Robust Design, Inverse Design

Research Question, Hypothesis, and New Knowledge in this Chapter

Research Question	RQ2b. What are the principles, mathematics, and method for designing adaptive strategies from desired future states of an E-CPSS under uncertainty?
Hypothesis	Hypothesis 2b. By using the Goal-Oriented Inverse Design Method to work backward from desired future states, enforce Design Capability Index constraints at every state transition, and update goals through feedback when observed performance deviates from projections, we produce adaptive strategies that remain robust across E-CPSS state transitions under uncertainty.
New Knowledge	New Knowledge 2b. A Goal-Oriented Inverse Design Method anchored in the principle of working backward from desired future states, with DCI-enforced transition conditions and feedback-driven goal updating, for deriving adaptive strategies across E-CPSS state transitions under uncertainty.

5.1 Frame of Reference

Building on the premise that policy designing occurs in a dynamic and interdependent environment, we hypothesize that policy designing can be modeled as an Evolving Cyber-Physical-Social System (E-CPSS) [1]. In such a system, policymakers combine physical, social, and cyber considerations when they formulate policies and evaluate policy performance over time [2,3]. By recognizing policy designing as an E-CPSS, we emphasize that effective policies must account for the constantly shifting interplay of these dimensions, rather than treating them as static or isolated components.

Within this perspective, we investigate the utility of a goal-oriented inverse designing method that offers a structured pathway for policymakers to begin with the desired long-term outcomes, be they reduced greenhouse gas emissions, balanced economic growth, or specific labor force targets, and work backward to identify feasible sets of decisions that realize these ends. While researchers have discussed this method extensively in engineering applications [4,5], its utility in policy design and open engineering is relatively unexplored. We contrast this approach with forward, single-objective optimization schemes, which often do not account for how policy relevance and effectiveness may degrade in evolving conditions. By prioritizing explicit goals from the outset and iterating backward to discover robust strategies, designers using the goal-oriented inverse design method can navigate trade-offs and uncertainties inherent in designing policy for complex, adaptive systems. An additional key element in this framing is our emphasis on adaptation. Policies must evolve (and so the designing) alongside changing stakeholder preferences, technological innovations, and external shocks (e.g., economic fluctuations or environmental disruptions). Even with sophisticated models, there will inevitably be gaps between expected and actual outcomes. By incorporating an explicit mechanism for adaptation, policymakers can periodically reassess goals and constraints, updating their decisions so that policies remain both viable and aligned with societal objectives over time.

The goal-oriented inverse design approach is characterized by several distinctive features that make it particularly suitable for policy design in E-CPSS. First, it is inherently normative and prescriptive; it begins with the specification of desired end-states, which aligns policymaking with long-term societal values and objectives rather than short-term political cycles. Second, it inverts the conventional cause-and-effect logic: instead of predicting outcomes from given interventions,

it identifies interventions from specified outcomes, thereby focusing design effort on feasibility and achievability under constraints. Third, we integrate the compromise decision support problem construct to naturally accommodate uncertainty by treating goals as fixed reference points while allowing the pathways to them to be flexible, adaptive, and revisable. Fourth, through the compromise decision problem construct designers satisfice across several criteria rather than optimize one, which they must do where economic growth, environmental sustainability, and social equity compete. We use these features to address a critical gap in policy design, and we offer a systematic and adaptive methodology that remains anchored to normative goals while being responsive to evolving system dynamics and uncertainties.

In what follows, we explore the foundational elements of designing policies as E-CPSS, examining (i) the integration of cyber, physical, and social dimensions in policy design (Section 5.1.1), (ii) the role of robustness in managing uncertainty (Section 5.1.2), and (iii) the significance of temporal considerations in adaptive policymaking (Section 5.1.3). In these subsections, we collectively lay the groundwork for designing policies using E-CPSS and the goal-oriented inverse designing method, and illustrate how adaptive, robust, and forward-looking strategies can be operationalized in policy contexts.

5.1.1 Designing Policies as Evolving Cyber Physical Social Systems

To design policies for dynamic, multifaceted challenges, designers must integrate physical, social, and computational dimensions. To this end, we hypothesize that **designing policies and industrial strategies can be modeled as an Evolving Cyber-Physical-Social System (E-CPSS) [1]**. This framing is particularly relevant to mechanical and systems engineers because it extends established design frameworks such as Design for X, Systems-of-Systems Engineering, and Model-Based Systems Engineering into the policy domain, where physical infrastructure, social dynamics, and computational analytics coevolve as coupled subsystems. In an E-CPSS framework, policymakers recognize that infrastructure and resources (physical), stakeholder behavior and collaboration (social), and analytics or computational intelligence (cyber) interact continuously over time. By capturing these interdependencies, we ensure that policies remain aligned with real-world complexities, rather than relying on static, siloed assumptions that quickly become outdated in rapidly changing contexts.

In this research, modeling policies through a goal-oriented inverse design method is a key step in addressing complex policy challenges and furthering the perspective of designing policies as E-CPSS. By beginning with explicit, future-focused objectives such as environmental quality targets, socioeconomic growth indicators, or community well-being benchmarks, we map backward to the interventions, decisions, and resource allocations required to achieve them. Unlike linear, top-down approaches that may overlook unforeseen factors, the goal-oriented inverse design method enables designers to accommodate uncertainty and changing conditions more naturally. It provides a structured yet flexible roadmap that allows policymakers to revise goals or policy pathways without abandoning the overarching vision.

To further enhance responsiveness, adaptation is crucial [6]. One of the defining traits of an E-CPSS is its dynamic nature; each subsystem (cyber, physical, and social) evolves in response to internal and external stimuli, creating new feedback loops and emergent behaviors that can render static policies ineffective rapidly. By embedding adaptation within the policy designing process, decision-makers acknowledge that current conditions and priorities may not hold true tomorrow [7]. Adaptive policy structures enable periodic or event-driven revaluation, triggered by economic disruptions, demographic shifts, or stakeholder feedback. In doing so, policymakers reduce the risk of policy obsolescence and ensure that, as the E-CPSS continues to evolve, policy objectives and strategies evolve alongside it. When designers build in adaptive capacity, they keep the policy relevant as conditions change, and stakeholders have grounds to trust that designers will revise commitments in the light of new evidence [8]. This adaptive embedding aligns closely with the principles of closed-loop design and design iteration under uncertainty that are well established in engineering design research. In this context, we operationalize adaptation by introducing measurable feedback variables and Design Capability Indices (DCIs) [9] that quantify how effectively a policy maintains performance under uncertainties. These quantitative metrics transform adaptation from a conceptual goal into an actionable design process, enabling policy designers to implement, monitor, and iteratively refine strategies with the same rigor applied in engineered systems.

An essential rationale for framing policy design as an Evolving Cyber-Physical-Social System (E-CPSS) is the recognition that each policy decision emerges from continual interactions among technological capabilities, physical infrastructure, and the evolving priorities of diverse stakeholders. By definition, E-CPSS captures dynamic feedback loops-such as those arising when

new data reveals a shift in economic or social trends, or when novel technologies reshape the feasibility of certain policy interventions. In this context, information and its structure in an information model become pivotal. Through a comprehensive information model, designers integrate inputs from real-time sensing (physical domain), algorithmic analytics (cyber domain), and human-driven goals or values (social domain), enabling stakeholders to engage in iterative, evidence-based deliberations. As policy requirements change or new insights come to light, the E-CPSS framework can adapt (update) the computational assumptions by revisiting physical constraints and realigning with social priorities. In doing so, the process of designing policies itself embodies the hallmarks of an E-CPSS: it remains dynamic, interactive, and capable of evolving in concert with the changing socio-technical environment that policies are meant to shape.

5.1.2 Towards Robustness in Designing Policy

The primary aspiration of every policy designer is to design policies and achieve solutions that are perceived as collective [10]. Stakeholders expect policies to be functional despite changes in the external environment, including physical and social domains, whether exogenous or endogenous [1]. To manage the uncertainty, policy designing should be capable of responding to "unexpected events and their impacts" [10] through mechanisms like 'learning', 'adaptation,' and 'improvisation' [10].

A fundamental consideration in designing policy for complex environments is robustness, the capacity of a system, institution, or policy to maintain functionality when confronted with significant shocks or disturbances [11]. In policymaking contexts, these "shocks" often arise from continuously evolving cyber, physical, and social domains, illustrating how a policy can be viewed as an Evolving Cyber-Physical-Social System (E-CPSS). For example, sudden technological breakthroughs (cyber), infrastructure disruptions or resource shortfalls (physical), and shifts in stakeholder priorities or demographics (social) may simultaneously reshape the environment in which a policy operates. Hence, effective decisions must be made in the face of unavoidable uncertainty [12,13], and modern policy design increasingly acknowledges uncertainty as an integral part of the long-term decision-making process [14].

Building on this acceptance of uncertainty, new modeling tools and decision-support methods have emerged that explicitly account for deep uncertainty, situations in which neither the precise nature of potential disruptions nor the best metrics for success are fully agreed upon [15, 16]. A prime

example is Robust Decision Making (RDM), an approach that generates a large ensemble of plausible futures (often by systematically varying uncertain parameters) to evaluate multiple candidate policies [17]. Through scenario discovery, policymakers can identify key vulnerabilities and opportunities for each policy, illuminating the circumstances in which it may fail or thrive [7, 18]. Iterative redesign then refines policies to enhance their overall robustness, ensuring they remain viable across numerous, often unpredictable states of the E-CPSS.

It is important, however, to distinguish robustness from adaptability. While adaptability refers to the processes by which institutions modify or reorganize in the face of new conditions, robustness is best viewed as a property that allows a system or policy to uphold core objectives amid disruptions and uncertainties. The ability to adapt depends on multiple contextual factors and the system's inherent robustness. In this chapter, we emphasize both robustness and adaptiveness, recognizing that although they serve different functions, they jointly enhance policy performance in dynamic E-CPSS environments.

The Goal-Oriented Inverse Design Method (GOIDM) proposed in this chapter explicitly incorporates both robustness and adaptability. In GOIDM, we embed robustness quantitatively through Design Capability Indices (DCIs), through which designers measure a policy's ability to maintain performance under variability and uncertainty. Simultaneously, GOIDM operationalizes adaptability through iterative feedback mechanisms that update policy goals and decisions based on observed deviations and changing conditions. Thus, GOIDM provides a structured approach to designing policies that are both robust (maintaining functionality amid shocks) and adaptive (evolving in response to new information).

Summarizing insights from Gilberto and Capano, policy robustness can be characterized by:

1. Maintaining the original problem definition and controlling agenda setting. The government plays a central role in articulating what the policy aims to address and in ensuring it remains salient [19, 20].
2. Redesigning policies coherently when negative effects arise. Robustness requires not only the capacity to steer the system politically but also an effective policy advisory system that can propose timely, integrative modifications [21].
3. Establishing policy procedures, information systems, and networks that offer prompt feedback. Early detection of potential misalignments, enabled by real-time data and

stakeholder engagement, allows policymakers to address emerging challenges before they escalate [22].

By embodying these characteristics, policymakers maintain a balance in robust policies between control, which preserves key objectives, flexibility, which modifies implementation details as conditions warrant, and agility, which enables rapid intervention or redirection when needed. We ground our approach in iterative modeling and scenario testing to align with the notion of an E-CPS, where policymakers must account for interlinked cyber, physical, and social domains evolving in tandem when designing robust policies. Ultimately, policymakers ensure that robust and adaptable policies preserve key objectives under uncertainty and adjust policies in response to new information and emergent complexities, thereby reinforcing long-term relevance and impact. To effectively manage uncertainty, policy designing must integrate adaptive mechanisms to allow policymakers to incorporate variability. The validity of early design decisions depends not only on adaptability but also on whether the expected improvement in design value outweighs the cost of uncertainty. Translating this insight into policymaking, we suggest that robust policies should be evaluated not only for their flexibility but also for whether their outcomes remain justified under epistemic uncertainty [23]. While we ensure that designed policies maintain reliability under variability, we propose a systematic method, GOIDM, to adapt them as the underlying system itself evolves. Through GOIDM, designers integrate robustness and adaptability when they combine quantitative robustness indices with iterative updates informed by feedback. Designers then revise policies as they encounter deep uncertainty and changing E-CPSS conditions.

5.1.3 Understanding the Temporal and Related Aspects

Policy Designing is a branch of policy sciences concerned with the study of policy processes [24]. Through policy designing, policymakers ground interventions in evidence and plan them deliberately. In an ideal setting, decision-makers would consistently adopt this approach, anchoring policy decisions in solid data and well-tested frameworks. Yet, in practice, policymaking often unfolds in a far more complex and chaotic manner, shaped by competing stakeholder interests, political pressures, and the inherent uncertainties of actual environments. This contrast highlights the gap between a theoretically streamlined design process and the multifaceted realities that influence how policies are conceived, negotiated, and enacted.

Adaptation has emerged as a central concern for policymakers grappling with rapidly evolving and uncertain conditions. In such settings, an effective policy not only strives to balance social, economic, and environmental objectives, but also must be robust, i.e., it can perform adequately across a wide range of possible futures, and adaptive, i.e., it can be reconfigured to address unforeseen developments [17]. Yet, how to systematically design policies with these qualities is rarely explored in the broader policy designing literature. As Walker and co-authors observe, much of the traditional applied work across engineering, social, and natural sciences treats uncertainty as a mere lack of information, leading to an overemphasis on data collection and processing to "reduce" uncertainties [10]. While more data can indeed refine forecasts, certain aspects of uncertainty may be irreducible, underscoring the need for new methods that leverage existing information but also enable policies to flex and evolve over time.

Scholars and policymakers have explored adaptive policymaking for nearly a century. In 1927, John Dewey [25] proposed viewing policies as experiments to foster continuous learning and recalibration based on real-world experience. Nonetheless, the rationale remains compelling: by embedding adaptability from the outset, treating policies as living structures that respond, iterate, and refine, the resilience and relevance of policy interventions can be markedly enhanced in an ever-changing world.

At the same time, forecasting future conditions and designing future states of the system [26] has become exceedingly difficult, prompting a surge of research into modeling and decision-support methods for deep uncertainty [27]. The ability to precisely foresee how situations will evolve, be they technological advances, sociopolitical shifts, or environmental events, remains elusive. In principle, if we could predict future states [26, 28] with high accuracy, we could evaluate each possible policy's outcomes and select the one that reliably maximizes benefits. However, this idealized scenario is seldom realistic in highly complex Evolving Cyber-Physical-Social Systems (E-CPSS). Frequently, the best-estimate models on which policymakers rely offer only limited insights into the interplay between ever-changing stakeholder priorities, emerging technologies, and infrastructural constraints. Consequently, static, one-shot policies designed using analyses fixed at a single point in time tend to be fragile when unexpected events inevitably emerge.

A more pragmatic approach is to embrace adaptive policies that can be iteratively updated as new information emerges and conditions change. In the face of deep uncertainty, goal-oriented design principles become particularly relevant. By defining long-term objectives first, then continuously

monitoring performance and feeding lessons back into the policymaking cycle, designers can recalibrate interventions in response to new data or stakeholder feedback. This adaptive mindset also recognizes that other actors, such as industry groups or community organizations, will adjust their strategies as policies change and new information emerges. Furthermore, attempting to design a single, all-encompassing solution for countless possible future risks, overlooking key vulnerabilities, or missing emerging opportunities. Instead, robust, adaptive policies seek to retain flexibility, allowing decision-makers to pivot strategy when critical assumptions are invalidated or when emergent windows of opportunity materialize. Ultimately, this forward-looking posture, grounded in E-CPSS thinking, prompts policymakers to evolve their policies rather than let them lag behind a highly uncertain world.

Acknowledging the temporal dimension of policymaking has become increasingly important, as policies rarely remain static; they must evolve to meet the changing realities of governance [29, 30]. Policies conceived at a particular point in time often require revisions when faced with unforeseen events, stakeholder realignments, or shifts in technology and infrastructure [31,32,33]. In other words, temporality underscores that policies are not permanent fixtures, but rather dynamic artifacts shaped by their socio-political and economic surroundings. Adopting an explicitly forward-looking and iterative approach to policy design helps practitioners identify potential points of failure and build the capacity to adapt without undermining core objectives [27]. Such temporal awareness naturally dovetails with the notion of Evolving Cyber-Physical-Social Systems (E-CPSS). As conditions change across these interlinked domains, the potential impacts on policy outcomes multiply [34]. Technological innovations (cyber) can alter resource dependencies (physical) or disrupt the balance of power among diverse interest groups (social), creating feedback loops that traditional static policy frameworks fail to capture [35]. By recognizing policy tools, stakeholder preferences, and infrastructural capacities all shift over time, designers can embed monitoring and adaptive features directly into the policy itself. This might involve phased rollouts, scenario-based triggers for reevaluation, or periodic stakeholder consultations, ultimately ensuring that, as the E-CPSS evolves, the policy remains relevant [36]. In this chapter, we focus on policy designing; however, the foundational principles outlined here extend beyond policymaking. They serve as key pillars in shaping business strategies for manufacturing enterprises, encompassing everything from material design to supply chain

management. Through these principles, enterprises can develop adaptive, robust strategies that enhance the effectiveness of adaptive policies in a dynamic environment.

From a goal-oriented design standpoint, temporal factors are integral to articulating clear end goals and structuring intermediary targets that can be adjusted as conditions evolve [11]. When policymakers begin by clarifying the long-term objectives, for instance, emission reduction thresholds or social equity benchmarks, they create a reference point against which interim achievements and emerging uncertainties can be measured [27]. This backward-mapping logic permits regular recalibration of strategies. Such recalibration becomes indispensable in turbulent contexts where deep uncertainty, such as unpredictable technological breakthroughs or abrupt economic downturns, compromises the utility of any one-off, static policy solution.

Ultimately, designing policies for a dynamic temporal landscape means embracing continual learning and adaptation. While a robust policy is equipped to stay on course amid disruptions, an adaptive policy flexibly adjusts its pathways to harness emerging opportunities or mitigate risks. Striking the right balance between these elements, robustness, and adaptability requires a systemic view of how policies fit within an ever-evolving E-CPSS. By integrating temporal analysis, feedback loops, and explicit adaptability into policy frameworks, decision-makers can move beyond short-lived fixes toward sustained, long-term governance success.

5.2 Understanding the Foundations of The Proposed Method

In this section, we introduce the core ideas and tools that underpin the proposed approach. Here, we will give a quick brief as to why it is essential to grasp these foundational elements, namely, the compromise Decision Support Problem, the different types of uncertainties, the role of design capability indices, and the utility of interpretable Self-Organizing Maps (iSOM) - to fully understand how the method addresses policy design challenges in evolving, uncertain environments.

5.2.1 The Compromise Decision Support Problem

The core of our proposed method revolves around the Compromise Decision Support Problem (cDSP). Understanding the underlying concept is essential for managing the trade-offs inherent in

designing policy, particularly in complex, evolving systems. The cDSP construct is rooted in decision theory and aims to provide a structured approach to navigating the inherent uncertainties and conflicts in decision-making [37]. Given that engineering design and policy development are fundamentally decision-making processes [38], the cDSP framework supports decision-making by balancing multiple, often conflicting, objectives while considering the system's long-term evolution. Unlike traditional optimization approaches, which focus on finding a single optimal solution, the cDSP construct uses a satisficing strategy to identify feasible, robust solutions that satisfy the necessary Karush–Kuhn–Tucker (KKT) conditions, even if they do not satisfy all sufficient conditions [39]. This perspective is particularly valuable when dealing with nonlinear, nonconvex objectives and constraints that are often present in socio-technical systems. As highlighted in the literature, satisficing allows designers to account for diverse and evolving stakeholder preferences, acknowledging that a single "optimal" solution may not be feasible or desirable in dynamic environments [39].

To implement the cDSP, we use a computational environment called DSIDES, and we manage trade-offs and evaluate alternative solutions systematically [40,41].

5.2.2 Modelling Uncertainties and Evaluation Metrics

As we mentioned earlier, in an evolving and uncertain environment, it is critical to understand the types of uncertainty and how to evaluate them to design a robust policy that enables policymakers to manage uncertainty effectively. Hence, a key aspect of the proposed method is the consideration of various types of uncertainties that affect decision-making in policy design.

In the robust concept exploration of materials, products, and associated manufacturing processes framework [42], the authors explore how to identify satisficing robust solutions in material design by considering three types of robust design, each corresponding to a specific source of uncertainty:

- Type I Robust Design: Focuses on achieving insensitivity to noise factors, which are external sources of variability, such as environmental changes or unforeseen events that can impact policy outcomes.
- Type II Robust Design: Aims to achieve robustness with respect to design/control variables, which may vary due to evolving stakeholder preferences, economic conditions, or technological developments.
- Type III Robust Design: Addresses uncertainty in the functional relationships or model

structure, where incomplete or imprecise knowledge about the relationship between design variables and system responses can lead to variation in outcomes.

To evaluate the effectiveness and robustness of the designed solution, we use the design capability indices (DCIs) and error margin indices (EMIs), as used in related work [43]. These metrics allow designers to assess how well a policy can accommodate uncertainties and evolving conditions. DCIs support Type I and Type II robust designs, while EMIs support Type III. In Section 5.4, for the test problem, we assume we only need to address Type I and Type II robust designs, so we use DCIs in our formulations. Although DCIs originated in engineering tolerance analysis, in this work they are used as abstract, dimensionless robustness metrics rather than physical tolerance measures. In the policy context, DCIs quantify the margin between achieved policy performance (e.g., GDP, electricity consumption, emissions) and policy-defined acceptable limits accounting for uncertainty. These limits represent normative or institutional thresholds, making DCIs directly transferable to social and policy system design. Understanding DCIs and EMIs helps policymakers compare alternative policy configurations and identify those that are more "robust" and "satisficing," rather than simply "optimal."

Variability in design variables is modeled using bounded uncertainty, and the resulting effect on system responses is estimated using a first-order (local linear) Taylor approximation around the nominal design point. This assumes that $f(x)$ is differentiable and that Δx_i are sufficiently small for higher-order and interaction terms to be neglected; uncertainty bounds are treated as decoupled, and correlations are not modeled. For a response function, f(x), the total response variation due to perturbations in the design vector $\mathrm{x} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{n}\}$ is given by:

$$\Delta Y = \sum_{i=1}^{n} \left| \frac{\partial f}{\partial x_i} \right| \cdot \Delta x_i \qquad (5.1)$$

where $\Delta \mathrm{x}_\mathrm{i}$ is the range of variation for design variable x_i derived from the defined uncertainty bounds. This allows us to estimate the expected spread in system performance. Depending on the nature of the system goal, the Design Capability Index (DCI) is computed as follows:

For "Larger-is-Better" objectives

$$DCI = \frac{(\mu_y - LRL)}{\Delta Y} \qquad (5.2)$$

For "Smaller-is-Better" objectives

$$DCI = \frac{(URL - \mu_y)}{\Delta Y} \qquad (5.3)$$

where μ_y is the mean response, LRL is the Lower Requirement Limit, and URL is the Upper Requirement Limit. A DCI value of 1 or greater indicates that the solution is robust against the modeled variability and meets the required performance threshold. Higher DCI values indicate greater robustness and safety margins under uncertainty. We apply this formulation to each system goal in Section 5.4 to evaluate policy robustness under Type I and II robust designs.

5.2.3 interpretable Self-Organizing Maps (iSOM)

iSOM is a tool for visualizing high-dimensional data through 2D plots. It is an unsupervised machine-learning algorithm, specifically an artificial neural network, and a modified form of the conventional Self-Organizing Map (SOM). Through the modification to the conventional SOM, designers are able to eliminate self-intersections, making iSOM plots inherently interpretable. Designers can apply iSOM to larger problems and can trace each region of the map back to input values. The plots generated using iSOM are valuable for visualizing the underlying relationships between input design variables and output responses [43]. By using iSOM, we can identify a set of common robust satisficing solutions that support designers at multiple levels, helping to manage different types of uncertainties [43]. Design preferences or targets are typically defined at the system level, while designers often want to understand regions of the design space across different stages that correspond to these targets.

Hence, a decision support framework that enables seamless navigation through high-dimensional design spaces is desirable. Recent work has proposed using the data-driven, interpretable self-organizing map (iSOM) as a visual enabler within decision support systems, facilitating design space exploration and understanding the trade-offs in system goals [44]. Using this framework, designers can modify design requirements easily and use iSOM plots to identify regions that meet new design preferences.

By leveraging iSOM to design robust policies for evolving cyber-physical-social systems, designers can navigate complex decision spaces where system goals, preferences, and uncertainties intersect. By using iSOM, we reveal relationships among various factors to identify robust solutions that adapt to changes in system parameters over time. This supports the designing of adaptive policies that evolve in response to changes and uncertainties.

5.3 Goal-Oriented Inverse Design Method (GOIDM) for Robust Policy Designing

In this section, we describe our method in detail and integrate elements from Sections 1 and 2. We explain the conceptual framework and features of our proposed method and demonstrate how we apply it to designing robust policies. We propose this method as a dynamic approach to policy design that accommodates uncertainties and evolving priorities.

In Figure 5.1, we demonstrate the conceptual framework of our Goal-Oriented Inverse Design Method (GOIDM) for adaptive policy design. On the X-axis, we represent different time states, from the current state to the target (final) state, while on the Y-axis, we capture goal attainment, representing how well system outcomes align with the desired long-term objectives.

At the "current state," both decision variables and goal values are known. This is the baseline for initiating the policy design process. The "target state" reflects the policy's aspirational endpoint, where specific economic, social, and environmental goals are to be met. Between the current and target states, we define multiple "transition states." These are intermediate time points that are used to progressively guide the system toward the final goal. For each transition state, we project goal values using statistical and forecasting techniques such as time-series analysis or compound annual growth models. These projected values serve as temporary benchmarks to evaluate progress over time.

To derive the decision variables for each transition state, we use the Compromise Decision Support Problem (cDSP) to compute satisficing solutions that yield policy decisions that may not be optimal but are robust under uncertainty and meet essential performance thresholds. The Design Capability Index (DCI) is used to evaluate the robustness of these solutions.

When the gap between a transition state and the final state is too large, or when adaptation demands finer control, designers can introduce additional transitional states. This flexibility allows for

gradual policy evolution and helps maintain feasibility across stages. One such transitional state is termed a conditional state. A conditional state is inserted only if prior deviations are significant and there is time to meaningfully adjust the trajectory. It is depicted in Figure 5.1 using a dotted line and a green dot, indicating its provisional and responsive nature. Furthermore, in Figure 5.1, we show the importance of feedback and information updates. At each state, policy designers compare achieved and projected goal values. When the design process underachieves or overachieves, they feed forward that information to the next state. Policymakers rely on continuous learning to recalibrate decisions and goal priorities as real-world conditions evolve. Consequently, they treat policy updates as part of an iterative, evolving process that maintains alignment with long-term objectives under deep uncertainty.

Through this structure, we demonstrate how policymakers and designers can use GOIDM to develop adaptive, feedback-driven, and robust policy pathways that effectively navigate the complexity of evolving Cyber-Physical-Social Systems (E-CPSS).

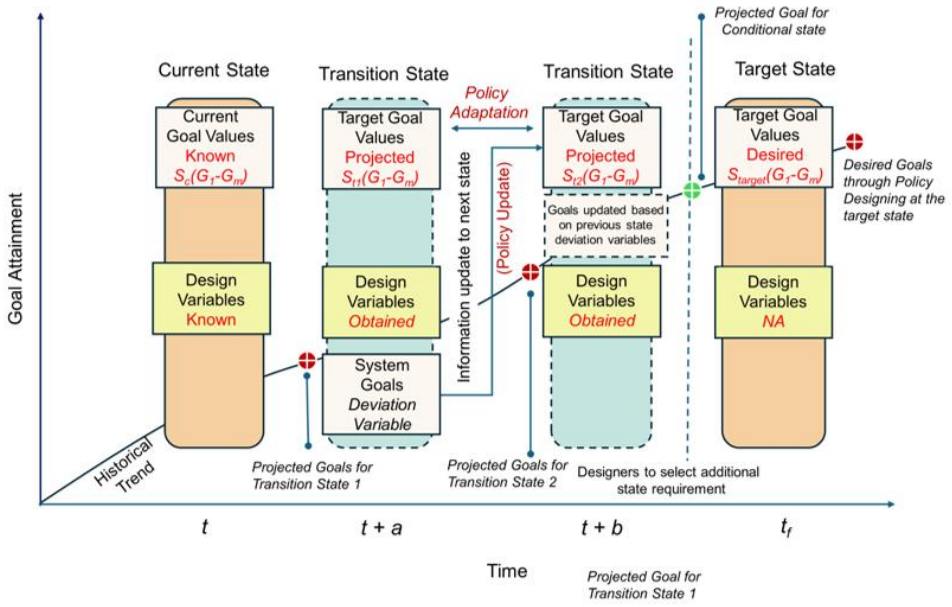


Figure 5.1: Conceptual Framework for Goal-Oriented Inverse Design

Policy designers who use traditional methods rely on incremental updates based on existing conditions. However, designers who use those methods often struggle to manage uncertainty and systemic change. In contrast, we use goal-oriented inverse design to work backward from desired long-term outcomes, systematically define transitional system states, and adapt policy variables accordingly. We present two figures to illustrate this approach. We present Figure 5.1 to

conceptually depict the approach at a high level of abstraction, and we present Figure 5.2 to illustrate how we apply backward and forward modeling to design policies.

In Figure 5.2, we illustrate the step-by-step implementation of the proposed Goal-Oriented Inverse Design Method (GOIDM), combining forward and backward modeling to design adaptive and robust policies. We include numbered labels to indicate the specific actions within the method. The process begins with forward modeling (Steps 1 to 3), in which policymakers define the current system state, transitional states, and the final target state. We use historical data and forecasting techniques to establish projected goals for each transitional state. We use these projected goals as future-oriented benchmarks to guide policy designers in developing adaptive policies.

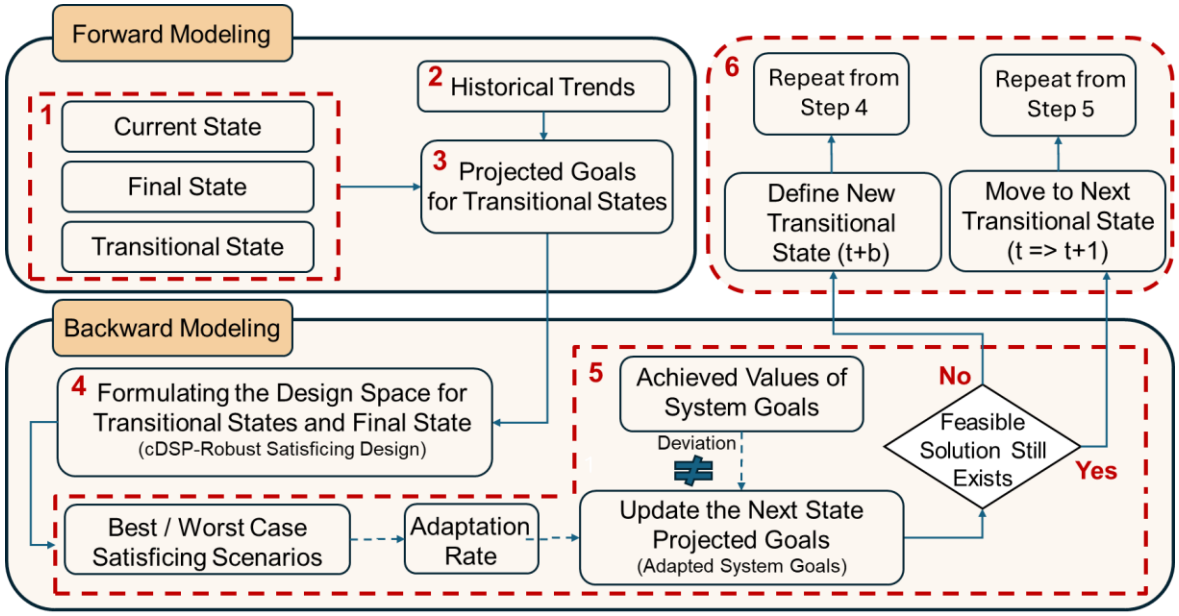


Figure 5.2: Forward and Backward Modeling in Goal-Oriented Inverse Design

Once goals are projected, we transition to backward modeling, where the design space is formulated using inverse design principles (Steps 4 and 5). At this stage, feasible policy decisions are identified by solving for the decision variables (e.g., labor allocations) that can achieve the desired goals. Based on the best and worst satisficing solutions, a dynamic adaptation rate is computed to quantify the flexibility available in meeting each goal. In Step 5, actual achieved values from implementation or simulation are compared to the projected goals. Any discrepancies are recorded as deviations. These deviations are then used to update the policy (system) goals for

subsequent states, ensuring that policies adapt over time to reflect real-world conditions. Finally, in Step 6, the revised goals are used to transition toward the final or next (new) transitional state (as shown in Figure 5.1), completing one cycle of adaptive policy design.

Through this iterative feedback loop, we enable the system to dynamically refine policy objectives and decision variables over time, ensuring robustness and alignment with long-term outcomes even in the face of uncertainty. The integrated forward-backward structure makes the GOIDM approach especially well-suited for designing policies such as Evolving Cyber-Physical-Social Systems, where continuous learning and adaptability are essential. In the following, we describe each step in detail. In Section 5.4, we implement the step-by-step process on our test problem, with reference to this section and Figure 5.2.

Step 1: Defining the Current, Final, and Transitional States of the System (forward modeling)

Policy evolution is modeled as a sequence of transitional states, each representing a discrete point in the policy development timeline. The goal is to transition from the current state to the desired target state while minimizing deviations and ensuring stability throughout the intermediate phases. This step should be defined from the designer's perspective, including the selection of appropriate time intervals between transitional states. These intervals reflect the system's capacity to monitor achieved goals and adapt projected goals for subsequent states accordingly.

Step 2: Defining Historical Trends (forward modeling)

The primary stage in this method involves establishing a baseline through the analysis of historical data. The system's current state is defined using already known policy variables. As shown in Figures 5.1 and 5.2, the variables and the goal values for the current (present state) are already known. Through the current state, designers can identify existing limitations and set realistic long-term goals.

Step 3: Projecting Future Goals and Designing Transitional States (forward modeling)

Instead of defining policies solely on current requirements, this step involves projecting future goals and setting intermediate milestones. The visual representation is shown in Figure 5.1. There can be multiple transitional states, here we explain the general significance of the transitional

states. Each transitional state is assigned a specific goal projection, formulated based on predicted deviations and external system changes. These projections are adjusted dynamically as the policy evolves. By projecting future goals and designing transitional states, designers can manage evolving systems through a structured process. Several projection methods can be used to estimate future values such as Linear Projection, Exponential Growth Models, Time Series Forecasting, or Machine Learning-Based Predictions. In this chapter, CAGR is used in the validation test problem as an illustrative example; while more accurate projections reduce the extent of required adaptation, the proposed framework is fundamentally designed for robustness and adaptive redesign even when projections are imperfect and system behavior deviates from expectations. These projected goals are considered system goal target values at each state.

Step 4: Formulating the Design Space Using Inverse Design Principles (backward modeling)
As shown in Step 4 of Figure 5.2, at each transitional state (an intermediate state), the projected goals are treated as target values, and a design space is constructed to identify a range of feasible policy options. To obtain the feasible policy options, we leverage the compromise Decision Problem Support Problem construct (detailed implementation in Section 5.4). This is specifically achieved through:

- Formulating the problem using the compromise Decision Support Problem construct.
- Establishing satisficing constraints that balance competing objectives.
- Introducing robustness constraints to account for uncertainty.

Step 5: Adaptive Policy Updates Through Feedback Mechanisms (backward modeling)
This step is labeled as Step 5 in Figure 5.2. To design an E-CPSS, a major criterion is to enable designers to incorporate feedback-driven design to account for events occurring in distinct time states. Through the proposed inverse policy designing method, we enable designers to employ feedback-driven adaptations, in which system goals are dynamically refined based on observed deviations from expected trajectories. These deviations are from the projected goals. For example, if a goal is underachieved by the end of Transitional State 1, this information is carried forward and used to update the projected goals for the next state (shown as Transitional State 2 in Figure 5.1) and further update the initially projected goals for Transitional State 2 to compensate for the underachievement in Transitional State 1. Additionally, an essential aspect of this approach is

updating goal weights based on how closely projected target values are met in each state. By using cDSP, goal preferences are dynamically adjusted through weight modifications to compensate for underachievement or overachievement of desired policy objectives, ensuring a balanced and effective policy design.

The adaptation rate for each goal is calculated backward, defined by the deviations between the best-and worst-case satisficing scenarios in the final state (See Equation 4). Starting from the last state (the Target State in Figure 5.1), satisficing solutions are analyzed to determine goal values for each scenario. The adaptation rate α_j for a Goal G_j is computed as:

$$a_j = \frac{\left|G_{j,best,\ t+1} - G_{j,worst,\ t+1}\right|}{G_{Target\ Projection(j)}} \qquad (5.4)$$

Where $G_{j,best}$ represents the value of the goal in the best satisficing scenario and represents the value of the goal in the worst satisficing scenario. This adaptation rate is then propagated backward to earlier states (for example, the Transitional State 2 in Figure 5.1) based on the deviation between the achieved and target goals. This ensures that goal-specific adjustments are dynamically applied in each state, allowing the adaptation process to account for discrepancies observed in prior decision periods.

A low adaptation rate means that if we do not achieve our desired goals in the transition state, we must fully adjust the target values for the goals in the next state. This is because, when adaptation is low, the difference between the worst and best scenarios, representing the lower and higher satisficing solutions compared to the target values of the prior state, is small. As a result, there is no viable way to bridge the gap between the target goal values and the achieved goals, even in the best-case scenario. Therefore, we need to update the designed goals based on Equation 5 and explore a new satisficing solution space. The system goal update equation we follow is:

$$G_j[t+1] = G_j[t+1] \pm (1 - \alpha_j) * (G_j[t] - G_j^{target\ projection}) \qquad (5.5)$$

Where (-) is used for maximization and (+) for minimization, and the last term represents the deviation, either d^- for underachievement or d^+ for overachievement. Through this formulation, we ensure that the adjustment process accounts for discrepancies observed in previous decision periods, refining goal-specific adaptations to maintain policy robustness.

Step 6: Iterative Refinement and Final Transition to the Target State

As shown in Step 6 in Figure 5.2, the transition from each intermediate transition state to the next follows an iterative improvement approach. With each iteration, new data and updated information are incorporated, ensuring the policy adapts to emerging uncertainties and environmental changes. This iterative refinement continues until the system successfully reaches the target state, where predefined long-term objectives are met. In the policy refinement process, two critical scenarios must be considered when assessing goal achievement in the final state. The first scenario arises when the goals are far from being achieved in the final state, as observed in the pre-final transitional state. In such cases, designers can introduce an additional transitional state into the analysis, allowing for a more gradual progression toward the final state. This approach is particularly relevant when there is a considerable time gap between the last transitional state and the final time state (i.e., when t and t+b differ significantly). As shown in Figure 5.1, this possibility is represented by the dotted line, with the projected goal for the conditional state denoted by the green dot. The second scenario occurs when the time state t+b is very close to the final state t_f, leaving minimal room for further adjustments. In such instances, designers may need to alert policymakers to the potential risk of failing to achieve the intended goals within the remaining timeframe. This proactive warning enables policymakers to take necessary corrective actions, such as adjusting expectations, reallocating resources, or implementing urgent policy measures, to mitigate deviations from desired outcomes.

5.4 Validation Through the Test Problem

To validate the proposed method, we use a test problem. For sustainable development, we require policymakers to manage complex trade-offs between economic, environmental, and social goals, making it an ideal case for our proposed method. In this context, policymakers must address uncertainties related to future resource availability, environmental impacts, and social equity. Through our method, we provide a structured yet flexible approach to designing policies that

balance these competing demands while maintaining robustness. By applying our method to sustainable development, we illustrate how policymakers accommodate uncertainties and the dynamic interplay among the cyber, physical, and social components of policy systems.

We demonstrate the utility of the proposed "Goal-Oriented Inverse Design for Robust Policy Design" method through a test example focused on sustainable development within the Persian Gulf region. This region is characterized by unique challenges, particularly rising energy consumption and the significant environmental impact of greenhouse gas (GHG) emissions. These issues align with global concerns, as increasing GHG emissions and rising temperatures have prompted international policy actions such as the Paris Agreement [45]. In recent decades, research has addressed these challenges using methods such as weighted goal programming and distributionally robust goal programming [46,47]. By selecting a country from this region and leveraging data from current sources and prior work, we show how policymakers can, by applying the proposed method, identify satisficing-robust solutions to address conflicting goals such as economic growth and environmental sustainability. We will outline the step-by-step application of the proposed method, demonstrating how designers can effectively address dynamic and uncertain conditions while achieving robust and adaptive policies.

While in the current test problem, we use linear models for demonstrating the applicability of the proposed goal-oriented inverse design method for policy development, using a well-known test problem from prior literature for comparison purposes, the proposed method is fully scalable to nonlinear, non-convex, and high-dimensional problems, as supported by prior applications of cDSP and iSOM in complex design formulation [26,39,43].

5.4.1 Problem Statement Data Source

The United Arab Emirates (UAE) is a key player in the Middle East, actively pursuing a more sustainable development path in recent years. However, the country faces significant challenges amid rapid economic growth. This accelerated development has increased demands on labor, infrastructure, and electricity consumption while contributing to rising greenhouse gas emissions [46]. Additionally, the UAE's electricity demand is growing faster than the global average, driven by its expanding economy and large-scale development projects. The country is also among the highest in the world in per capita energy consumption, highlighting the urgent need for sustainable energy strategies. In this chapter, we address the problem of sustainable development under

uncertainty in design variable variability, arising from increasing concerns about the sustainability of the economy, environment, energy, and society. A key aspect of this problem is determining a robust satisficing labor allocation across eight key economic sectors to ensure sustainable development while accounting for uncertainty and multiple criteria. The eight sectors are Agriculture, Crude oil, Natural Gas and Mining, Manufacturing and Electricity, Construction and Real Estate, Trade and Transport, Restaurants and Hotels, Banking and Financial Services, Government, and Social and Personal Services. In this problem, we use data collected from previous studies [46,47] and from the World Bank [48]. In this test problem, we consider two key uncertainties and robust design in sustainable development policy design. Type I robust design arises from external influences, such as economic fluctuations and pandemics (e.g., COVID-19), which introduce unpredictability into sustainability goals. Type II robust design addresses variability in workforce distribution across economic sectors, modeled with a 4% fluctuation over six years based on historical trends [48]. Therefore, to manage these uncertainties, we use design capability indices, as explained in Section 5.2.2. In the following, we formulate the problem step by step and derive the policies using the forward and backward modeling approach in the proposed method, as shown in Figure 5.2.

5.4.2 Establish Forward Modelling of the Goal-Oriented, Inverse Method

In this section, we define each state, explain how to establish forward modeling to guide the transition from the current state to the final future state, and align our content with forward modeling as described in Steps 1 to 3 of Figure 5.2 and Section 5.3.

5.4.2.1 Defining Current, Final, and Transitional States of the System

This step corresponds to Step 1 in Section 5.3, where we define the current, transitional, and final states as the foundation for forward modeling based on historical data. To demonstrate the efficiency of the proposed method, we assume that our current state is the same as the original work, which was in 2017, and the target state is set for 2030. Selecting 2030 as the final state aligns with the United Nations Agenda for Sustainable Development, which outlines 17 Sustainable Development Goals (SDGs) to be achieved by 2030 [49]. These goals include ensuring access to affordable, reliable, and clean energy, promoting sustained economic growth with full productive employment, and taking urgent action to mitigate environmental degradation. We define 2023 as

the “Transitional State” since it represents the most up-to-date data available at the time of writing and falls at the midpoint of the defined period (2017-2030). The target values are set as follows: GDP at 3,110,323 AED (817.7 billion USD), electricity consumption at 361,513 GWh, greenhouse gas (GHG) emissions at 286,051 kilotons, and the labor force at 9,115,735 persons.

5.4.2.2 Defining Historical Trends

By referring to various valid data sources (Step 2 in Section 3 and Figure 5.2), we obtain historical trends for our four defined goals: GDP, electricity consumption, GHG emissions, and the number of employees, up to the “Current State” (the year 2017). We present all historical data from 2000 to 2017 in Figure 5.3. GDP growth data is obtained from the World Bank database [48]. Electricity consumption data is sourced from the International Energy Agency (IEA) [50]. Greenhouse gas (GHG) emissions data is retrieved from Our World in Data [51], and employment is gathered from the World Bank [48].

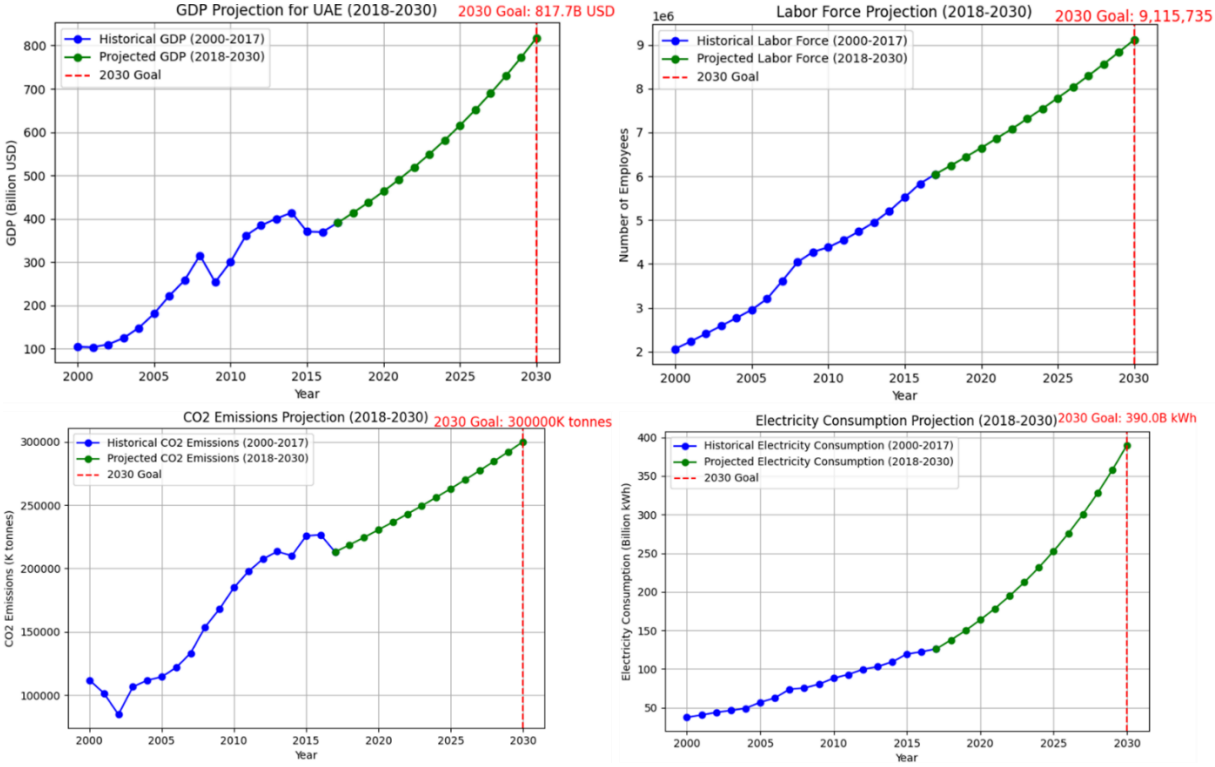


Figure 5.3: Historical Trends (2000-2017) and Projected Future Values (2018-2030)

5.4.2.3 Projecting Future Goals and Designing Transitional States

In this section (corresponding to Step 3 in Section 5.3), we complete forward modeling in Figure 5.2 by projecting future goals and assigning target values to transitional states using historical data. To project future values, we use the Compound Annual Growth Rate (CAGR) method, in which a consistent growth rate over time is assumed. We emphasize that CAGR is used here only as an illustrative projection mechanism for the test problem, and any domain-specific projection model may be substituted without changing the structure of the proposed method. Using this method, we provide a smooth, exponential trend to estimate future values for our defined goals. The formula for CAGR is:

$$CAGR = \left(V_f / V_i\right)^{(1/n)} - 1 \qquad (5.6)$$

where:

- V_f is the target value (GDP, electricity consumption, GHG emissions, and number of employees in 2030).
- V_I is the last known historical value (The value in 2017).
- n is the number of years between the last known value and the target year (2030 – 2017 = 13 years).

Once the growth rate is determined, future values are projected using:

$$V_t = V_{(t-1)} \times (1 + CAGR) \qquad (5.7)$$

where each year's value builds on the previous year's value. The projections are visualized with historical trends for clarity in Figure 5.3. Since CAGR is used for projection, some goal trajectories may appear approximately linear; however, these projections are illustrative. Smaller deviations imply less adaptation, while larger deviations require greater adaptive redesign, which is the focus of this work.

Now, regarding the projected goals, we estimate the target values for the transition state for 2023.

Table 5.1: Projected Values for the Goals In the Transitional State (Year 2023*)*

Criteria	Year 2023 projected values
GDP	2,015,768 (in AED)
Electricity Consumption	212,340 (in GWh)
GHG Emission	249,422 (in kiloton)
Number of Employees	7,307,789

5.4.3 Formulating the Design Space Using Inverse Design Principles

In this section, we present the mathematical formulation, computational results, and robust solution space by applying the proposed method to the sustainable development test problem. This aligns with Step 4 in Section 5.3 and Figure 5.2, where projected goals are used to construct a design space through backward modeling. These results include visualizations of the solution space, key trade-offs between economic and environmental goals, and insights into the robustness of different policy alternatives. We also highlight the sensitivity of the results to various uncertainties associated with the system. Using these computational results, designers can leverage the practical applicability of the Goal-Oriented Inverse Design method to support robust policy decisions.

5.4.3.1 Mathematical Formulation for the Final State (Year 2030)

These four sustainability criteria are defined based on the problem statement and prior studies [46,47], which represent the key performance metrics for the UAE's sustainable development (GDP, electricity consumption, GHG emissions, and employment) and will be the target goals in our inverse design formulation.

Goal 1 = Gross domestic product – GDP - (in local currency)

Goal 2 = Electricity consumption (in GWh)

Goal 3 = Greenhouse gas – GHG - emissions (in K Tones)

Goal 4 = Number of employees (in thousands)

The decision variable X_i represent labor allocation across eight major economic sectors in the UAE. These sectors are selected based on national accounts and economic reports used in [46,47] and will be the system variables used in the cDSP formulation:

X1 = Agriculture

X2 = Crude oil, Natural Gas and Mining

X3 = Manufacturing and Electricity

X4 = Construction and Real Estate

X5 = Trade and Transport

X6 = Restaurants and Hotels

X7 = Banking and Financial Services

X8 = Government, Social and Personal Services

In this study, an average variability of 4% is considered for the decision variables over a period of six years, reflecting the annual mean percentage of population fluctuations observed in different periods. This assumption is based on historical trends. The minimum values and variability of each decision variable are presented in Table 5.2. These variability values are then used to compute response variations (ΔY) through the formulation outlined in Section 5.2.2 and are integral to the calculation of DCIs.

Table 5.2: System Variables, Ranges, and Variability (No. of Employees)

Sr. No	Decision Variable (X_i)	Minimum Value	Variability (Δx_i)
1	X1	258867	±10355
2	X2	74284	±2971
3	X3	687686	±27507
4	X4	1505931	±60237
5	X5	1403509	±56140
6	X6	236357	±9454
7	X7	81037	±3241
8	X8	810366	±32415

The following goal models (Equations 8–11) are used in the cDSP formulation to achieve a robust design solution for allocating employees across economic sectors in 2030. These models are based on econometric relationships identified in prior studies [46,47]:

Model for GDP

$$GDP = 0.035705X1 + 4.762018X2 + 0.183853X3 + 0.085018X4 + 0.179355X5 + 0.082073X6 + 1.06595X7 + 0.09702X8 \quad (5.8)$$

Model for Electricity Consumption

$$Electricity\ consumption = 0.005358X1 + 0.066171X2 + 0.028007X3 + 0.020969X4 + 0.018068X5 + 0.008266X6 + 0.162398X7 + 0.009761X8 \quad (5.9)$$

Model for GHG Emission

$$GHG\ emission = 0.018356X1 + 1.823233X2 + 0.07039X3 + 0.002837X4 + 0.006663X5 + 0.002741X6 + 0.035564X7 + 0.003239X8 \quad (5.10)$$

Model for Number of Employees

$$Number\ of\ employees = X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 \quad (5.11)$$

Next, we formulate the problem in cDSP construct. As mentioned in Section 5.2.2, a DCI value of 1 or higher indicates that the solution is robust to the modeled variability and meets the required performance threshold, as defined by our system constraint. The target values for DCI are designer-defined and reflect the desired robustness level for each goal. Higher DCI values indicate greater robustness and safety margins under uncertainty; therefore, our system goals are formulated to maximize DCI values. The mathematical formulation of the problem, including DCI goals (derived by converting system goals in Equations 8–11 into DCI form using Equations 1–3) and constraints to achieve robust Design Types I and II, is presented as follows:

Given

DCI $_{Target}$ for GDP goal = 2, DCI $_{Target}$ for GHG Emissions goal = 2, DCI $_{Target}$ for Electricity consumption goal = 3, and DCI $_{Target}$ for Number of employees goal = 3.

Find

System Variables

X1: X8

Deviations from the target of GDP

$d1^-$: Underachievement of GDP

$d1^+$: Overachievement of GDP

Deviations from the target of GHG Emissions

$d2^-$: Emissions below the limit

$d2^+$ Excess GHG emissions above the limit

Deviations from the target of Electricity consumption

$d3^-$: Underconsumption of electricity

$d3^+$: Overconsumption of electricity

Deviations from the target of the Number of employees

$d4^-$: Underemployment

$d4^+$: Overemployment

Satisfy

System Constraints

- Robust solution constraint for GDP:

$DCI_{GDP}(x) \geq 1$

Robust solution constraint for GHG Emissions:

$DCI_{GHG\ Emissions}(x) \geq 1$

- Robust solution constraint for Electricity consumption:

$DCI_{Electricity\ consumption}(x) \geq 1$

- Robust solution constraint for the Number of employees:

$DCI_{Number\ of\ employees}(x) \geq 1$

- Limitation for the maximum number of employees:

$X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 \leq 9{,}800{,}000$

System Goals

Goal 1:

- Maximize DCI for GDP:

$$DCI_{GDP}(x) / DCI_{Target,\ GDP} + d1^- - d1^+ = 1$$

- Maximize DCI for Electricity consumption:

$\text{DCI}_{\text{EC}}(x) / \text{DCI}_{\text{Target, EC}} + d2^- - d2^+ = 1$

- Maximize DCI for GHG Emissions:

$\text{DCI}_{\text{GHG Emissions}}(x) / \text{DCI}_{\text{Target, GHG Emissions}} + d3^- - d3^+ = 1$

- Maximize DCI for Number of Employees:

$\text{DCI}_{\text{Number of employees}}(x) / \text{DCI}_{\text{Target, Number of employees}} + d4^- - d4^+ = 1$

Where:

$$DCI = \frac{\mu_y - LRL}{\Delta Y} \text{ or } DCI = \frac{URL - \mu_y}{\Delta Y}$$

$$\Delta Y = \sum_{i=1}^{n} \left|\frac{\partial f}{\partial x_i}\right| \cdot \Delta x_i$$

Bounds on system variables

Difined in Table 5.2.

Bounds on deviation variables

$$d1^+ \geq 0 \ , d1^- \geq 0 \ , d2^+ \geq 0, d2^- \geq 0 \ , d3^+ \geq 0 \ , d3^-, d4^+ \geq 0 \ , d4^- \geq 0$$

Minimize

We minimize the deviation function

$$Z = \sum_{i=1}^{4} W_i (d_i^- + d_i^+), \quad \sum_{i=1}^{4} W_i = 1$$

To calculate the DCI goals, we first determine the response variations for each goal in Equation 12. Then, we set the target values defined in Section 5.4.2.1 as the lower requirement limit for goals we aim to maximize and as the upper requirement limit for goals we aim to minimize, ensuring that performance stays within the desired range and avoiding underachievement or excessive overperformance.

$$\Delta Y_{GDP} \approx 42140,\ \Delta Y_{EC} \approx 4219,\ \Delta Y_{GHGE} \approx 8334,$$
$$\Delta Y_{NoE} = 202320 \qquad (12)$$

5.4.3.2 Computational Results and Robust Design Exploration for the Final State (Year 2030)

We begin the inverse decision-based robust design exploration from the final state, Year 2030. The cDSP for this final state is formulated with DCI goals that capture all the criteria requirements for the end policy: the distribution of employees across sectors. In our test problem, we use 40 scenarios, each representing a different way to assign weights. In each scenario, we consider specific logic for each goal. Some scenarios and their corresponding interpretations are shown in Table 5.3. By considering different weight assignments, we demonstrate how different stakeholders or decision-makers may approach the problem based on their preferences. Additionally, depending on the scenario, we can adapt the policy based on the actual values achieved for our goals in the "transitional states". This will be discussed further in Section 5.4.4. By running the cDSP formulation in DSIDES [40], we obtain the results for each scenario (see Table 5.3).

After generating robust design solutions through the cDSP formulation, we visualize and explore the robust solution spaces across different goals and weight assignments. To achieve this, the weights assigned to different design scenarios and their corresponding goal values are used to train the iSOM algorithm. We then generate four iSOM plots for the four goals and four additional plots for the four weights, as shown in Figures 4 and 5. We use iSOM plots to analyze the relationships between input weights (W1–W4) and the corresponding cDSP goals. For example, when W1 or W4 exceeds 0.7, electricity consumption tends to be high, while GHG emissions remain low. Conversely, a high value of W2 or W3 results in lower GDP. Additionally, using iSOM plots

provides insights into interactions among the cDSP goals. For example, high GDP values correspond to low GHG emissions but high electricity consumption.

Table 5.3: Sample Scenarios with Weight Assignments and Corresponding Results for The Final State (The Year 2030)

Scenario #	GDP (w_1)	Elec. (w_2)	Emission (w_3)	Employees (w_4)	Interpretation	Goal 1 DCI	Goal 2 DCI	Goal 3 DCI	Goal 4 DCI
1	0.25	0.25	0.25	0.25	Equal weight to all goals	1.62	2.00	2.66	3.00
2	0.80	0.05	0.05	0.1	Strong focus on GDP	2.00	2.00	1.84	3.00
3	0.05	0.80	0.05	0.1	Strong focus on Electricity Reduction	1.62	2.00	2.66	3.00
4	0.05	0.05	0.8	0.1	Strong focus on Emission Reduction	1.00	2.00	2.82	3.00
5	0.05	0.10	0.1	0.75	Strong focus on Labor Force Numbers	1.62	2.00	2.66	3.00
…	…	…	…	…	…	…	…	…	…
40	0.35	0.25	0.15	0.25	GDP slightly prioritized over Employees	2.00	2.00	1.84	3.00

By determining the satisficing limits, that is, acceptable values for each goal, we identify robust satisficing solution spaces (nodes) on the iSOM plots for each goal. As designers, we select greater robustness against uncertainties by choosing spaces with higher DCI values. Therefore, the acceptable values for the final state are as follows: Goal 1 (DCI GDP) $\geq$ 1.5, Goal 2 (DCI Electricity consumption) $\geq$ 1.5, Goal 3 (DCI GHG emissions) $\geq$ 1.8, Goal 4 (DCI Number of employees) $\geq$ 2.5.

We determine the acceptable spaces by outlining the selected nodes with red hexagons. The red dots indicate samples mapped to the nodes. The size of the red dot increases based on the number of hits that a node (space) has. A larger number of hits implies that more samples were mapped to that particular node. By considering all acceptable nodes, 15 iSOM grid nodes are identified as the common satisficing region for all goals, as depicted by the black nodes in Figure 5.6. Black cells represent iSOM grid points where all goals are simultaneously satisfied. These regions are determined based on the training data hits (red dots) shown in Figure 5.5. In this satisficing solution

space, nodes with hits correspond to 24 solution points. The results associated with two of these solution points, with their corresponding weight interpretations, are summarized in Table 5.4.

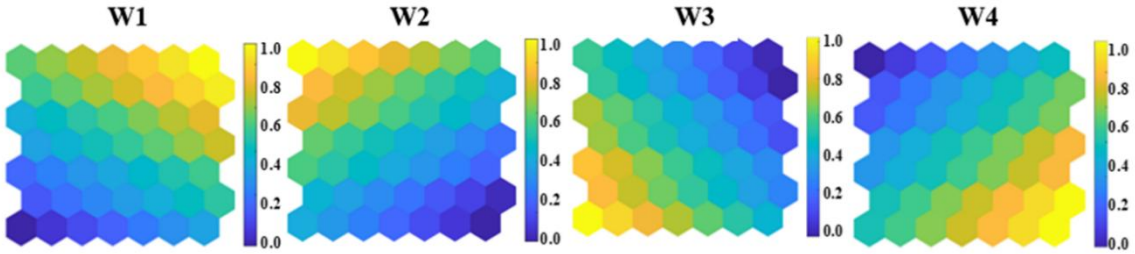


Figure 5.4: iSOM Plots For All Weights (Scenarios)

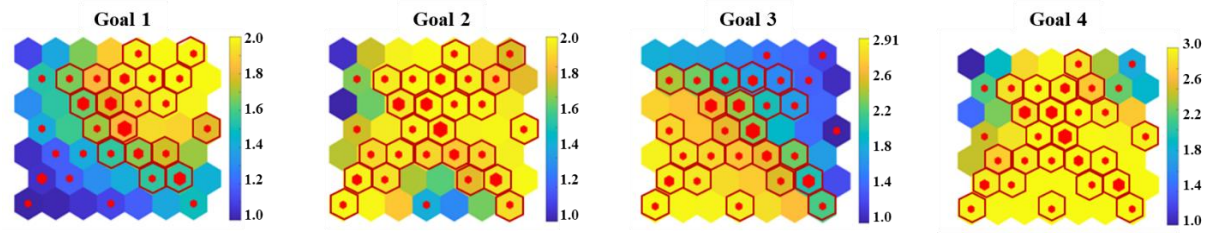


Figure 5.5: iSOM Plots for All Goals

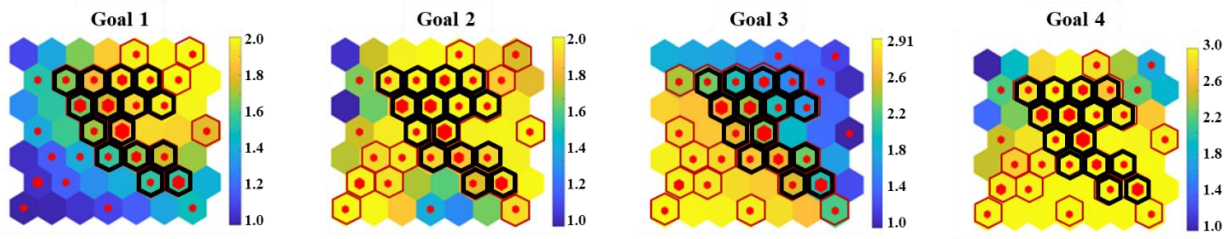


Figure 5.6: The Black iSOM Grid Points Indicate the Common Satisficing Solution Region for All Goals

Table 5.4: Sample Scenarios and Robust Solution Points for the Final State

Scenario	Interpretation	X1	X2	X3	X4	X5	X6	X7	X8
1	Equal weight to all goals	258867	74284	687686	1505931	4739559	236357	1409645	810366
2	Strong focus on GDP	258867	78048	687686	1505931	4737049	236357	1408391	810366

By considering all design solutions, we can develop a policy for long-term labor force allocation in each sector (Year 2030) and adapt the design requirements based on the achieved goals in the transition states. The adaptation policy is presented in Section 5.4.4. Using Equation 4, we calculate the adaptation rate for each goal in Year 2030, based on the worst- and best-case scenarios, and the target values for Year 2023 (Transition State). The adaptation rate for each goal is shown in Table 5.5. By using the adaptation rate, we dynamically adjust policy goals based on deviations from expected outcomes, ensuring continuous refinement and robustness in the face of uncertainty.

Table 5.5: Adaptation Rate Calculation Between the Years 2030 and 2023

	Goal 1	Goal 2	Goal 3	Goal 4
Best Scenario	3084280	381562	275777	9722695
Worst Scenario	3042140	385781	291666	9318055
Adaptation Rate	0.029	0.020	0.064	0.055

Next, we analyze the transition state (Year 2023) as a short-term reference point to track progress toward desired goals in the intermediate years. By studying the achieved goals in the transition state, designers can adapt the policy and update the system goals, along with their corresponding weights, to address any gaps in specific goals for the next state.

5.4.3.3 Mathematical Formulation for the Transition State (Year 2023)

Since in Section 5.4.2.1 we defined Year 2023 as the transition state to analyze the achieved goals, we formulate the problem in cDSP similarly to Section 5.4.3.1, with the difference that the target values for each goal are estimated based on the projected data as follows: GDP - 2,015,768, Electricity Consumption - 212,340, GHG Emission - 249,422, Number of Employees - 7,307,789. The minimum values for system variables and their variabilities are provided in Table 5.6. Therefore, the goal variability can be calculated as follows: $\Delta Y_{GDP} \approx 30782$, $\Delta Y_{EC} \approx 3082$, $\Delta Y_{GHGE} \approx 6081$, $\Delta Y_{NoE} \approx 162194$.

Table 5.6: System Variables, Ranges, And Variability For State 2023 (No. Of Employees)

Sr. No	Decision Variable (Xi)	Minimum Value	Variability (Δx_i)
1	X1	207525	±8301
2	X2	59551	±2382
3	X3	551296	±22051
4	X4	1207256	±48290
5	X5	1125148	±45006
6	X6	189480	±7579
7	X7	64965	±2598
8	X8	649644	±25986

We define the DCI target values as follows: DCI $_{Target}$ for GDP goal = 2, DCI $_{Target}$ for GHG Emissions goal = 2, DCI $_{Target}$ for Electricity consumption goal = 2, and DCI $_{Target}$ for Number of employees goal = 4. Additionally, we set a maximum limit of 8,300,000 employees based on the projected data.

5.4.3.4 Computational Results and Robust Design Exploration for the Transition State (Year 2023)

For the transition state (Year 2023), we follow the same process as in Year 2030 to formulate the cDSP, generate design solutions, and explore robust satisficing spaces. By formulating the problem in DSIDES, we find out there are no feasible solutions that satisfy all DCI constraints, necessitating the relaxation of the constraint bounds. This means that while we can obtain satisficing solutions for long-term goals and the final state, the feasibility of solutions for short-term goals and mid-term states is not achievable. We interpret this result as evidence that the team cannot satisfy robustness constraints with the current short-term targets. To address this issue, we use the adaptation rate, as defined in Equation 4, to relax system goals and achieve an acceptable, robust feasible solution for the transition states.

$$r_j = 1 \pm a_j \quad (13)$$

Where (-) is used for maximization and (+) for minimization. With reference to Equation 13, we decrease the target values for goals we aim to maximize and increase the target values for goals we aim to minimize. By applying this relaxation while accounting for the adaptation rate, we can obtain a possible satisficing solution that remains aligned with the worst-case scenario for the

next state. This ensures that a satisficing solution will still be achievable in the following state. The adjusted goals are shown in Table 5.7.

Table 5.7: Relaxation Rates And Adjusted Goal Values For Feasibility

	Goal 1	**Goal 2**	**Goal 3**	**Goal 4**
Relaxation Rate	%79.1	%120	%106.4	%94.5
Relaxed Goals	1973628	216559	265311	7307789

Using the same assigned weights (scenarios) and adjusted goal values, we apply the iSOM algorithm to analyze relationships among the design criteria. By selecting nodes with acceptable DCI values, DCI for GDP goal = 1.5, DCI for GHG Emissions goal = 2, DCI for Electricity consumption goal = 1.5, and DCI for Number of employees goal = 4, we find the robust solutions as shown in Figure 5.7. The results for the two solution points of the transition state, along with their corresponding weight interpretations, are summarized in Table 5.8.

With the solutions for both the transition state (Year 2023) and the final state (Year 2030), designers can now design a policy at the current state (Year 2017) and track the achieved goals to adapt the design solution by monitoring the transition states. In the next section, we demonstrate how to implement adaptive policy updates to ensure a robust, adaptive design.

5.4.4 Adaptive Policy Updates Through Feedback Mechanisms

Here, we map the problem to Step 5 in Figure 5.2 and Section 5.3, which represent feedback-driven policy adjustments based on system evolution and adaptation rates. Our goal in this step is to enable designers to create adaptive policies. Adaptive policy updates must be structured within an iterative feedback loop, where observed deviations from projected goals inform refinements in real time. This ensures that policies dynamically evolve in response to both expected and unexpected environmental changes, solidifying their role within the E-CPSS. In this section, we collect data from Year 2023 for the studied criteria - GDP, electricity consumption, GHG emissions, and number of employees - to evaluate the achieved system goals.

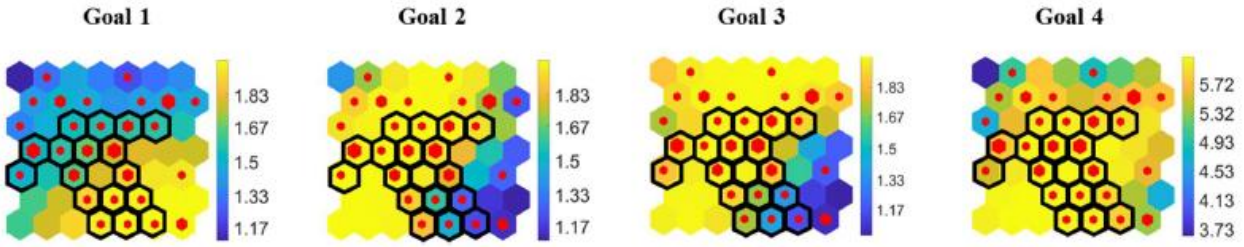


Figure 5.7: Satisficing Solution Space for State 2023

We then compare these results with the targeted goals defined in the designed policy as part of the robust solution. The values achieved are summarized in Table 5.9.

Table 5.8: Sample Scenarios and Robust Solution Points for Transition State 2023

Scenario	Interpretation	X1	X2	X3	X4	X5	X6	X7	X8
1	Equal weight to all goals	207525	86107	551296	1207256	5012388	189480	396304	649644
27	Focus on GDP and Employee	207525	89632	551296	1207256	4654293	189480	453686	649644

Table 5.9: Targeted and Achieved System Goals for The Year 2023

Criteria	Year 2023 targeted values	Year 2023 achieved values
GDP (AED)	1,973,628	1,860,925
Electricity Consumption (GWh)	216,559	158,000
GHG Emission (kiloton)	265,311	229,384
Number of Employees (Labor Force)	7,307,789	6,793,436

By comparing the achieved goals with the targeted goals for this state, we observe that we did not reach the GDP and labor force targets we projected, which changes what designers must achieve by 2030. On the other hand, the results for electricity consumption and GHG emissions are better than expected.

For the next step, we need to update the system goals and assign weights for the next state using Equation 5. Through this process, we adjust the priority of underachieved or overachieved goals in Table 5.10 to adapt the designed policy by updating the system goals for the (next) final state.

Table 5.10: Updated System Goals and Weights for the Final State (Year 2030)

Criteria	Year 2030 targeted goals values	Year 2030 updated goals	Updated Weights
GDP (AED)	3,000,000	3,089,148	Increase ↑
Electricity Consumption (GWh)	390,000	447,395	Decrease ↓
GHG Emission (kiloton)	360,000	393,638	Decrease ↓
Number of Employees	9,115,735	9,601,799	Increase ↑

Based on the proposed method, there are two possible outcomes for the updated next state. First, if a scenario satisfies the newly updated goals and weights, we transition to it, prioritizing those with higher gaps. However, if no scenario meets the newly defined goals, an intermediate transitional state must be introduced to bridge the existing gaps and ensure gradual progression toward the final state. The second outcome is discussed in Section 5.4.5. In our case, however, it is possible to adjust the goals and weights to achieve a satisficing scenario in the next state.

Using the proposed method, the designer can easily modify the design requirements and use iSOM plots to identify regions that meet new design preferences. In our case, the design requirement is to achieve high DCI values for GDP and the number of employees, and lower values for electricity consumption and GHG emissions. In this case, the designer can easily identify the desired regions on the iSOM plots in Figure 5.5, indicated by the yellow color on the Goal 1 and Goal 4 plots. These regions correspond to high weights on GDP (W1) and the number of employees (W4), and to low weights on electricity consumption (W2) and GHG emissions (W3), as shown in Figure 5.8. Thus, the designer can quickly make informed decisions about weight combinations using iSOM plots to develop an adapted policy. This makes iSOM more suitable than other dimensionality reduction techniques, which often lack the interpretability and inverse-mapping capabilities needed for multi-objective design tasks. The ability to visually link goals to input weights supports both forward (input-to-output) and inverse (output-to-input) mapping, aligning well with the scenario-based structure of cDSP and the characteristics of goal-oriented inverse design. By using iSOM, designers can explore trade-offs under uncertainty, identify satisficing regions, and quickly adapt policies based on evolving system goals.

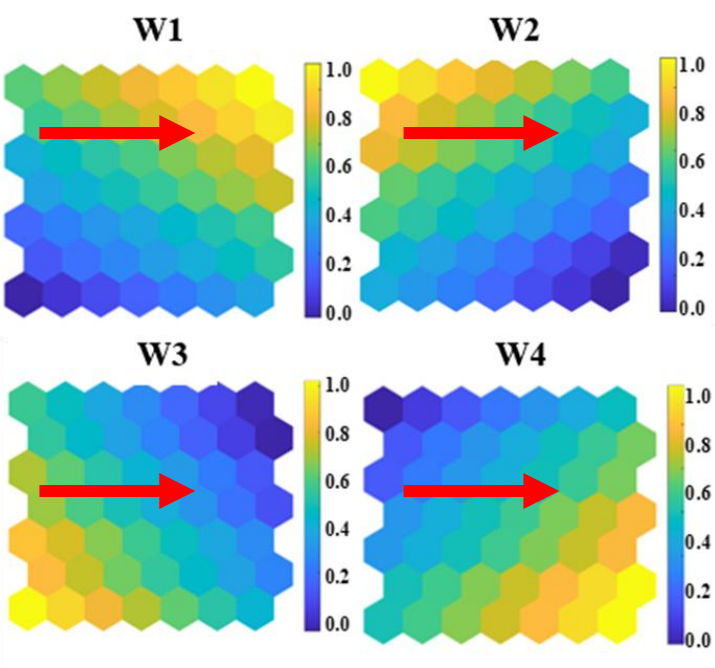


Figure 5.8: Isom Plot, Updating the Weights

We present the updated scenario selection based on adjusted design preferences in Table 5.11. Scenario 28 prioritizes electricity consumption and GHG emissions, resulting in lower GDP and labor force values. Conversely, Scenario 27 focuses more on GDP and employment, increasing their values while keeping electricity consumption and GHG emissions within a satisficing range. Through this process, we design an adapted policy that aligns with the updated system goals.

Table 5.11: Scenario Adjustments for Adaptive Policy Design

Scenario #	Weight Interpretation	Goal 1 (AED)	Goal 2 (GWh)	Goal 3 (kilotn)	Goal 4 (No of employees)
28	Focus more on Electricity and GHG Emission	3042139	381561	276694	9342327

27	Focus more on GDP and Number of Employees	30892 79	3815 61	29166 6	9722695

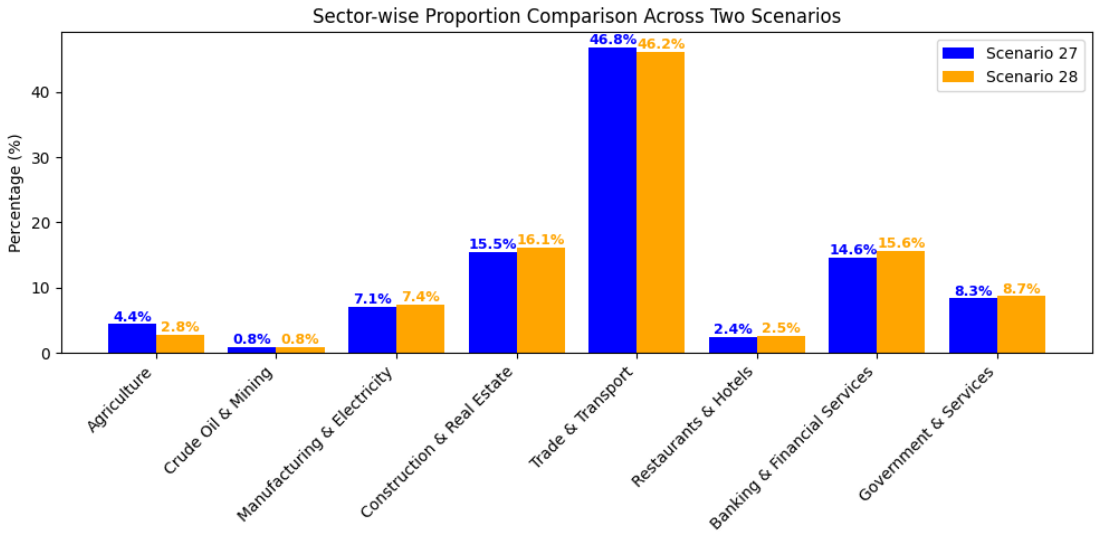


Figure 5.9: Sector-Wise Employment Distribution Across Scenarios 27 and 28

In Figure 5.9, we illustrate the differences in sector-wise employment distribution between Scenarios 27 and 28. Scenario 27, which emphasizes GDP and the number of employees, results in higher employment in sectors like Trade and Transport and Agriculture. In contrast, Scenario 28, which prioritizes reducing electricity consumption and GHG emissions, shows a decline in employment within these sectors while slightly increasing employment in Banking and Financial Services. Through this method, we show the development of an adapted labor allocation policy in response to evolving system goals.

By adjusting the distribution of employment in the "Banking and Financial Services", "Trade and Transport", "Crude Oil and Mining" and "Agriculture" sectors, we design a policy that is not only robust for the next state but also responds effectively to deviations in the transition state to cover existing gaps. The updated policy is presented in Table 5.12. In this table, we compare the employment patterns that decision makers obtain when they assign different priorities across sectors. Scenario 27 assigns more workers to labor- and energy-intensive sectors to support GDP and jobs, while Scenario 28 shifts employment toward less energy-intensive sectors to reduce electricity use and GHG emissions.

The process of adapting the policy based on the achieved values continues iteratively until the final state is reached, when multiple transition states exist. Following the proposed method outlined in Sections 3.4 and 3.5, if no satisficing solution is found given the updated system goals and weights, we need to introduce a new transition state to ensure a feasible progression toward the final state. This is discussed in Section 5.4.5, where we introduce a new transition state (Year 2027) to enable gradual progression toward the final state.

Table 5.12: Comparison of Employment Distribution by Sector in Scenarios 27 and 28 (No. of Employees)

Sector	Employees Scenario 27	Employees Scenario 28
Agriculture	426,335	258,867
Crude Oil and Mining	80,609	74,284
Manufacturing and Electricity	687,686	687,686
Construction and Real Estate	1,505,931	1,505,931
Trade and Transport	4,553,126	4,311,575
Restaurants and Hotels	236,357	236,357
Banking and Financial Services	1,422,285	1,457,262
Government and Services	810,366	810,366

5.4.5 Iterative Refinement and Final Transition to the Target State

In this section, we complete the method shown in Step 6 of Section 5.3 and in Figure 5.2, where the designer uses iterative refinements to adapt the policy toward the final state or introduce a new transitional state if needed. If no satisficing solution is found after adjusting the goals and their weights, two critical scenarios must be considered when evaluating goal achievement in the final state.

First Scenario, if the goals are far from being achieved in the final state, as seen from the pre-final transitional state, designers can introduce an additional transitional state. This allows for a more gradual progression toward the final state. Second Scenario, if the timeline is close to the final state, leaving little room for further adjustments, designers should alert policymakers to the risk of not achieving the intended goals within the remaining timeframe.

This proactive warning allows policymakers to take necessary corrective actions. Here, we assume that there is no satisficing space at State “The Year 2030”. Therefore, we set 2027 as a transitional

year and update the target goals based on the identified gaps and the defined adaptation rates in Table 5.13.

Table 5.13: Updated Goals for Transitional State (Year 2027)

Criteria	Year 2027 Projected Goals	Year 2027 Updated Goals
GDP	2,530,431	2,641,257
Electricity Consumption	300,544	358,281
GHG Emission	277,176	311,044
Number of Employees (Labor force allocation)	8,291,754	8,781,006

To obtain the mathematical formulation and then derive the results and robust design solutions, we follow the same process outlined in Section 5.4.3. Regarding the DCI formulation, we define the target DCI goals as follows: DCI_{Target} for GDP goal = 2, DCI_{Target} for GHG Emissions goal = 1, DCI_{Target} for Electricity consumption goal = 1 and DCI_{Target} for Number of employees goal = 3. By selecting nodes with acceptable DCI values - DCI for GDP goal = 2, DCI for GHG emissions goal = 1, DCI for electricity consumption goal = 1, and DCI for number of employees goal = 2 - we identify the robust solutions, as shown in Figure 5.10.

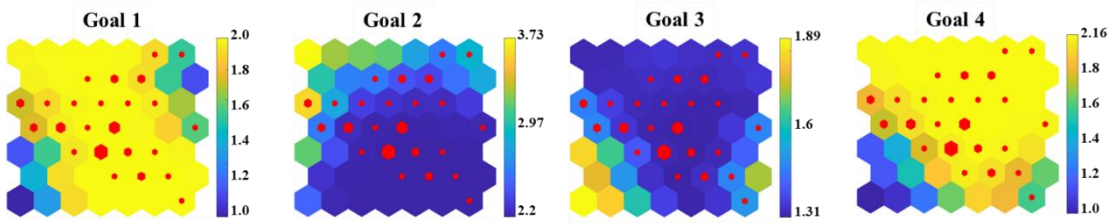


Figure 5.10: Satisficing Solution Space for Transition State Year 2027

These acceptable DCI values are chosen because, in this transitional state, we need to prioritize nodes with higher DCI values for Goals 1 and 4, which did not meet their 2023 targets. This means we prioritize these two goals and select the regions with higher weights for them.

Based on the obtained satisficing solution, we observe that all scenarios either yield unacceptable values (DCI for Goals 1 and 2 below 2) or result in a single labor allocation. Hence, since we have very strict constraints for this transition state - because we designed this new transition state to address the gaps existing in the previous state - policymakers should be very careful and precise in designing this solution. The only solution points for the Year 2027 transition state are

summarized in Table 5.14. Decision variables X1-X8 represent sector-wise employment levels (number of employees), and G.1-G.4 denote the corresponding DCIs for GDP, electricity consumption, GHG emissions, and total employment, respectively.

By implementing the policy based on the robust satisficing solution for this transition state, selecting a scenario that focuses on GDP and the number of employees for the final state, and monitoring the results for Year 2027, we can gradually progress toward the Year 2030 targets. If, in 2027, the goals achieved do not meet our expectations, we will iterate the process outlined in Section 5.4.5. This means either defining a new transition state or alerting policymakers to the risk of failing to achieve the intended goals within the remaining timeframe.

Table 5.14: Robust Solution (Number Of Employees) For the Transition State Year 2027

X1	X2	X3	X4	X5	X6	X7	X8	G.1	G.2	G.3	G.4
2755473	74284	687686	1505931	1403509	236357	1526394	810366	2.00	2.21	1.31	2.16

5.4.6 Synthesizing the Approach: GOIDM as E-CPSS

We adopt the Goal-Oriented Inverse Design Method for policy design because it fundamentally addresses the core challenge of policymaking in dynamic, interconnected environments: maintaining commitment to long-term objectives while remaining responsive to evolving conditions. Traditional policy design often relies on forward-looking models that project current trends into the future, but these approaches struggle when system dynamics are nonlinear, uncertain, or subject to unexpected disruptions. In GOIDM, we invert this logic by starting with explicit long-term goals and working backward to identify feasible interventions, making it inherently prescriptive and goal-driven rather than merely predictive.

Modeling policy design as an Evolving Cyber-Physical-Social System (E-CPSS) recognizes that modern policies operate in environments where technological capabilities (cyber), physical infrastructure and resources (physical), and human behaviors and institutional structures (social) continuously interact and co-evolve. In such systems, static policies become obsolete as these domains shift. GOIDM operationalizes this E-CPSS perspective by:

1. **Integrating multiple domains:** Through this method, we explicitly incorporate cyber (data analytics, forecasting models), physical (resource constraints, infrastructure), and social (employment, stakeholder preferences) dimensions throughout the design process.
2. **Embracing evolution:** By defining transition states and enabling iterative adaptation, GOIDM treats policies not as fixed interventions but as evolving entities that co-develop with the system they aim to steer.
3. **Quantifying robustness across domains:** Design Capability Indices (DCIs) provide cross-domain metrics that measure how well policies withstand variability in all three domains simultaneously.

Through the sustainable development test problem, we examine how designers use GOIDM to retain long term goals while revising intermediate targets when observed or assumed conditions differ from earlier projections. Through the case study, we show that policymakers can systematically transition from their current state to the target state while incorporating real-world feedback and uncertainty. The method's strength lies in its structured yet flexible approach: it has clear pathways (via cDSP construct) while allowing course corrections (through adaptive updates) when conditions change or projections prove inaccurate.

Through GOIDM, designers use DCI values to evaluate robustness at each specified state and revise intermediate goals when new observations differ from prior projections. This is particularly valuable in E-CPSS contexts, where policies must remain effective amid technological disruptions, social shifts, and evolving physical constraints. Here, robustness does not mean that a policy is immune to all uncertainty. Instead, it means the policy can remain within acceptable limits as conditions change. In this context, DCIs measure the extent of the buffer between achieved outcomes and these limits, helping policymakers understand how adaptive a policy is when facing uncertainty or unexpected deviations.

Comparison with Traditional Approaches

To evaluate GOIDM's effectiveness, in Table 5.15, we compare it with traditional policy design approaches, the weighted goal programming and distributionally robust goal programming methods from prior studies [46,47] that addressed the same sustainable development challenge. While both approaches aim for the same 2030 sustainability targets, their methodologies and outcomes differ significantly.

Table 5.15: Comparison of GOIDM With Traditional Goal Programming–Based Policy Design Approaches.

Aspect	Traditional Goal Programming [46,47]	GOIDM (This Study)	Key Difference
Design Philosophy	Forward-looking, predictive optimization	Goal-oriented, inverse design	GOIDM starts from the desired ends and works backward
Uncertainty Handling	Stochastic programming with probability distributions	Robust design with DCIs and explicit variability bounds	GOIDM quantifies robustness against specific uncertainty types
Temporal Dimension	Single-period optimization (2030 only)	Multi-period with transition states (2023, 2027, 2030)	GOIDM breaks long-term goals into achievable intermediate steps
Adaptation Mechanism	None (static solution)	Systematic adaptation based on feedback and deviations	GOIDM enables mid-course corrections as conditions evolve
Solution Space	Single "optimal" allocation	Satisficing solution space with multiple robust options	GOIDM provides policymakers with flexibility in implementation
Visualization and Exploration	Limited to sensitivity analysis	iSOM-enabled design space visualization and inverse mapping	GOIDM supports intuitive exploration of trade-offs and scenarios
2023 Transition State	Not considered	Explicitly designed and evaluated (92.3% GDP, 74.4% electricity of targets achieved)	GOIDM provides early warning of deviations from the trajectory
2030 Target Adjustment	Fixed targets	Dynamically updated based on transition state performance	GOIDM realistically adapts long-term goals based on feasibility

Advantages of GOIDM Over Traditional Approaches

Through the comparison we observe several advantages of GOIDM over traditional policy design methods:

1. **Proactive Adaptation:** While traditional methods produce static policies that may fail when conditions change, GOIDM embeds adaptation directly into the design process. Through the systematic feedback mechanism (Equation 5.5), we allow policies to evolve based on observed performance, preventing the policy obsolescence common in rapidly changing E-CPSS environments.

2. **Explicit Robustness Quantification:** Traditional approaches often treat uncertainty through probability distributions or scenario analysis. GOIDM operationalizes robustness through Design Capability Indices (DCIs) that directly measure a policy's ability to maintain performance under specific variability (e.g., ±4% labor fluctuations). This provides policymakers with tangible metrics for comparing policy alternatives.
3. **Transition State Management:** GOIDM recognizes that long-term goals (e.g., 2030 SDGs) must be approached through feasible intermediate steps. By explicitly designing and evaluating transition states (2023), GOIDM provides early warning of implementation challenges—such as the 25.6% shortfall in achieving the electricity target—enabling timely adjustments before deviations become irreversible.
4. **Structured Yet Flexible Exploration:** Traditional optimization typically converges to a single "optimal" solution, which may be fragile to model assumptions. GOIDM, through its satisficing approach and iSOM-enabled visualization, identifies regions of robust solutions, giving policymakers multiple viable options with clear trade-offs.
5. **E-CPSS Integration:** GOIDM naturally accommodates the interconnectedness of the cyber, physical, and social domains. In our case study, the method simultaneously addresses cyber (forecasting models, iSOM analytics), physical (energy infrastructure, emissions), and social (employment, sectoral preferences) dimensions, ensuring policies remain relevant as these domains co-evolve.
6. **Practical Implementation Ability:** By breaking ambitious long-term goals into achievable transition states and providing clear adaptation triggers, through GOIDM, we transform visionary objectives into actionable policy pathways. By this we bridge common gap between policy aspiration and implementation that plagues many sustainable development initiatives.
7. In essence, GOIDM does not merely improve upon traditional policy design methods; it reorients the entire process around the realities of evolving complex systems. By treating policies as adaptive components of E-CPSS rather than static interventions, GOIDM enables policymakers to navigate deep uncertainty while remaining committed to long-term societal goals. This approach is particularly valuable in contexts like sustainable development, where success requires balancing competing objectives across multiple domains amid constantly changing conditions.

5.5 Closing Remarks for Chapter 5

In this chapter, we address the principal question posed in the abstract: *How can policymakers design policies as evolving Cyber-Physical-Social Systems (E-CPSS) that remain effective amid dynamic conditions?* We answer this question by presenting goal-oriented inverse design as a foundational method for policymakers and designers to design robust and adaptive policies (Sections 3 and 4). We demonstrate how adaptive goal-oriented inverse design, in concert with quantitative robustness measures via Design Capability Indices, equips policymakers with a structured pathway to navigate uncertainty and evolving system states.

Key Takeaways

Through this work, we present several actionable insights for policy design in complex, evolving systems. Policymakers can use GOIDM to (i) anchor policy design in explicit long-term goals, then systematically work backward to identify feasible interventions; (ii) quantify robustness explicitly via Design Capability Indices, enabling comparison of policy alternatives under uncertainty; and (iii) embed adaptive feedback mechanisms that allow policies to be refined as conditions change or new data become available. Through these steps, we ensure policies remain aligned with desired outcomes without sacrificing responsiveness to emerging realities.

Actionable Insights for Policymakers

We encourage policymakers to adopt a goal-first mindset, beginning with clear, measurable long-term outcomes rather than immediate constraints. Using the GOIDM framework, practitioners can define intermediate transition states, project future targets using historical data, and employ satisficing strategies to identify robust, flexible policy options. Incorporating tools such as iSOM for visualization and DCIs for robustness scoring can help teams communicate trade-offs and make informed, adaptive decisions as systems evolve. By treating policies as living, learning components of an E-CPSS, decision-makers can maintain alignment with societal goals while remaining agile in implementation.

In Section 5.3, we lay the foundation for designing E-CPSS policies by integrating the cyber, physical, and social dimensions. Mechanical engineers play an important role in working toward a sustainable future. To align with these goals, in Section 5.4 we demonstrate the utility of our method on a sustainable development test problem, illustrating how policymakers systematically

transition from current to target states while ensuring robustness. By bridging short- and long-term objectives (Section 5.4.2), managing uncertainty through Design Capability Indices (Section 5.4.3), and implementing feedback-driven policy adaptations (Section 5.4.4), we propose a framework that policymakers can use to achieve robust policy design. We demonstrate that policymakers can sustain policy effectiveness amid uncertainty and drive iterative refinements, keeping policies viable over time.

Beyond policy design, we suggest that practitioners in manufacturing, urban planning, healthcare administration, energy systems, and supply chain management can use this approach to devise adaptive strategies under uncertain demand, shifting stakeholder priorities, and evolving technological landscapes. By establishing long-term targets, defining intermediate operational states, and adapting strategies based on real-time feedback, practitioners can enhance decision-making under uncertainty and system evolution.

Limitations: In GOIDM framework, we currently require predefined uncertainty bounds and assume stationary variability over time, whereas real-world E-CPSS often exhibit dynamic, endogenous uncertainties that evolve with the system itself. This presents new opportunities in systems engineering and operations research for designing complex systems. Additionally, scalability to high-dimensional, multi-actor policy environments remains a computational challenge, pointing toward opportunities for reduced-order modeling, AI-assisted scenario generation, and decentralized inverse design protocols. Rather than imposing constraints, we open pathways to extend GOIDM into more adaptive, robust, and computationally efficient implementations in increasingly complex socio-technical systems.

Open Research Issues

We suggest several promising avenues for future research as follows**.** (1) dynamic uncertainty modeling, where variability bounds themselves evolve over time, requiring extensions of DCI formulations to handle time-dependent robustness; (2) multi-level goal reconciliation, developing methods to formally link high-level normative targets with operational sub-goals across scales in decentralized governance contexts; (3) AI-enhanced adaptation, leveraging reinforcement learning and generative models for automated scenario discovery, real-time policy adjustment, and predictive robustness assessment in high-dimensional E-CPSS; and (4) validation in new policy domains such as public health crisis response or climate adaptation, where deep uncertainty, rapid change, and competing values test the limits of adaptive, goal-oriented design. Advancing these

areas will extend GOIDM's theoretical foundations and practical utility in increasingly complex socio-technical systems.

We opened this chapter by setting out to address RQ2b and to examine Hypothesis 2b. Within the scope of the UAE sustainable development problem we study, spanning three temporal states from 2017 through 2030, the evidence we present is consistent with that hypothesis: working backward from desired future states with the Goal-Oriented Inverse Design Method, enforcing Design Capability Index constraints at each state transition rather than only at the endpoint, and updating goals through feedback when observed performance deviates from projection, yields adaptive strategies that remain robust across the transitions we model. On that basis we offer the new knowledge we set out to produce: a Goal-Oriented Inverse Design Method, anchored in the principle of working backward from desired future states with DCI-enforced transition conditions and feedback-driven goal updating, for deriving adaptive strategies across E-CPSS state transitions under uncertainty. We take this to contribute $D(s_i \rightarrow s_{i+1})$, the inter-state decision map, which together with $D(s_i)$ from Chapter 4 completes the Decision Layer of the DSIM. With respect to verification and validation, we read the Frame of Reference as offering theoretical structural validity, the UAE problem as a reasonable vehicle for empirical structural validity given the length and complexity of its design horizon, and the demonstrated cross-transition robustness as empirical performance validity within the bounds of this single problem; theoretical performance validity we defer to Chapter 9.

With the Decision Layer in place, we can read what a system is doing, structure decisions within it, and design pathways through its transitions. Two pressures now arise against the limits of these methods: as goals multiply, as the decision space expands to dozens of coupled variables, and as scenarios grow too numerous for iSOM visualization to support unaided navigation, the satisficing philosophy remains sound but our own cognitive capacity as designers does not keep pace. That is the challenge we take up in Chapter 6, where we augment the designer's intelligence with structured generative AI rather than replace it.

References

[1] Bhalerao, M. J., 2024, "On Designing Public Policy as an Evolving Cyber-Physical-Social System," M.S. thesis, University of Oklahoma, Norman, OK, USA. https://shareok.org/handle/11244/340292

[2] Pasandideh, S., Pereira, P., and Gomes, L., 2022, "Cyber-Physical-Social Systems: Taxonomy, Challenges and Opportunities," *IEEE Access*, 10, pp. 42404–42419. https://doi.org/10.1109/ACCESS.2022.3167441

[3] Gupta, H., Bhalerao, M. J., Honeycutt, W. T., Allen, J. K., and Mistree, F., 2025, "On Designing Evolving Cyber–Physical–Social Systems: A Decision-Based Design Perspective," *ASME Journal of Computing and Information Science in Engineering*, 25(12), p. 120803. https://doi.org/10.1115/1.4070094

[4] Nellippallil, A. B., Rangaraj, V., Gautham, B. P., Singh, A. K., Allen, J. K., and Mistree, F., 2018, "An Inverse, Decision-Based Design Method for Integrated Design Exploration of Materials, Products, and Manufacturing Processes," *ASME Journal of Mechanical Design*, 140(11), p. 111403. https://doi.org/10.1115/1.4041050

[5] Zunger, A., 2022, "Inverse Design in Search of Materials With Target Functionalities," *Nature Reviews Chemistry*, 2(4), p. 0121. https://doi.org/10.1038/s41570-018-0121

[6] Dovers, S. R., and Hezri, A. A., 2010, "Institutions and Policy Processes: The Means to the Ends of Adaptation," *Wiley Interdisciplinary Reviews: Climate Change*, 1(2), pp. 212–231.

[7] Gorddard, R., Colloff, M. J., Wise, R. M., Ware, D., and Dunlop, M., 2016, "Values, Rules and Knowledge: Adaptation as Change in the Decision Context," *Environmental Science and Policy*, 57, pp. 60–69.

[8] Miranda, S., and Caruso, R., 2023, "The Importance of Transparency in Regulation," *ASME Open Journal of Engineering*, 2, p. 021002.

[9] Nellippallil, A. B., Mohan, P., Allen, J. K., and Mistree, F., 2020, "An Inverse, Decision-Based Design Method for Robust Concept Exploration," *ASME Journal of Mechanical Design*, 142(8), p. 081703.

[10] Capano, G., and Woo, J. J., 2018, "Designing Policy Robustness: Outputs and Processes," *Policy and Society*, 37(4), pp. 422–440. https://doi.org/10.1080/14494035.2018.1504494

[11] Mens, M. J. P., Klijn, F., de Bruijn, K. M., and van Beek, E., 2011, "The Meaning of System Robustness for Flood Risk Management," *Environmental Science and Policy*, 14(8), pp. 1121–1131.

[12] Dessai, S., Hulme, M., Lempert, R., and Pielke, R., 2009, "Do We Need Better Predictions to Adapt to a Changing Climate?" *Eos, Transactions American Geophysical Union*, 90(13), pp. 111–112.

[13] Lempert, R. J., Popper, S., and Bankes, S., 2003, *Shaping the Next One Hundred Years: New Methods for Quantitative, Long-Term Policy Analysis*, RAND Corporation, Santa Monica, CA, USA.

[14] Walker, W. E., Haasnoot, M., and Kwakkel, J. H., 2013, "Adapt or Perish: A Review of Planning Approaches for Adaptation Under Deep Uncertainty," *Sustainability*, 5(3), pp. 955–979.

[15] Hamarat, C., Kwakkel, J. H., and Pruyt, E., 2013, "Adaptive Robust Design Under Deep Uncertainty," *Technological Forecasting and Social Change*, 80(3), pp. 408–418.

[16] Lempert, R. J., and Collins, M. T., 2007, "Managing the Risk of Uncertain Threshold Responses: Comparison of Robust, Optimum, and Precautionary Approaches," *Risk Analysis: An International Journal*, 27(4), pp. 1009–1026.

[17] Kwakkel, J. H., and Pruyt, E., 2013, "Exploratory Modelling and Analysis: An Approach for Model-Based Foresight Under Deep Uncertainty," *Technological Forecasting and Social Change*, 80(3), pp. 419–431.

[18] Torres, E. S., Thies, P. R., and Lawless, M., 2023, "Offshore Logistics: Scenario Planning and Installation Modelling of Floating Offshore Wind Projects," *ASME Open Journal of Engineering*, 2, p. 021013.

[19] Rasch, E., and Tsebelis, G., eds., 2011, *The Role of Governments in Legislative Agenda Setting*, Routledge, London, UK.

[20] Green-Pedersen, C., and Walgrave, S., eds., 2020, *Agenda Setting, Policies, and Political Systems: A Comparative Approach*, University of Chicago Press, Chicago, IL, USA.

[21] Craft, J., and Howlett, M., 2012, "Policy Formulation, Governance Shifts and Policy Influence: Location and Content in Policy Advisory Systems," *Journal of Public Policy*, 32(2), pp. 79–98.

[22] Willis, C. D., Riley, B. L., Best, A., and Ongolo-Zogo, P., 2012, "Strengthening Health Systems Through Networks: The Need for Measurement and Feedback," *Health Policy and Planning*, 27(Suppl. 4), pp. iv62–iv66.

[23] Hulse, D., Hoyle, C., Tumer, I. Y., and Goebel, K., 2021, "How Uncertain Is Too Uncertain? Validity Tests for Early Resilient and Risk-Based Design Processes," *ASME Journal of Mechanical Design*, 143(1), p. 011702.

[24] Howlett, M., and Mukherjee, I., 2018, "Introduction: The Importance of Policy Design: Effective Processes, Tools and Outcomes," *Routledge Handbook of Policy Design*, M. Howlett and I. Mukherjee, eds., Routledge, New York, NY, USA, pp. 3–19.

[25] Dewey, J., 1927, *The Public and Its Problems*, Holt and Company, New York, NY, USA.

[26] Mandegari, S., Bhalerao, M. J., Allen, J. K., and Mistree, F., 2024, "Predicting Future States of Evolving Systems," *ASME Computers and Information in Engineering Conference*, Washington, DC, USA, Paper No. DETC2024-146340.

[27] Marchau, V. A., Walker, W. E., Bloemen, P. J., and Popper, S. W., 2019, *Decision Making Under Deep Uncertainty: From Theory to Practice*, Springer Nature, 405.

[28] Haynes, J. P. E., Bhalerao, M. J., Honeycutt, W. T., Allen, J. K., and Mistree, F., 2024, "Predictive Modeling for Public Policy Design: The Impact of Artificial Lights at Night (ALAN) on Bird Strikes," *ASME Computers and Information in Engineering Conference*, Washington, DC, USA, Paper No. DETC2024-143780.

[29] Howlett, M., Mukherjee, I., and Rayner, J., 2018, "Understanding Policy Designs Over Time," *Routledge Handbook of Policy Design*, M. Howlett and I. Mukherjee, eds., Routledge, London, UK, pp. 136–144.

[30] Sabatier, P. A., 1988, "An Advocacy Coalition Framework of Policy Change and the Role of Policy-Oriented Learning Therein," *Policy Sciences*, 21(2–3), pp. 129–168.

[31] Lasswell, H. D., 1956, *The Decision Process: Seven Categories of Functional Analysis*, University of Maryland Press, College Park, MD, USA.

[32] Bobrow, D. B., 2006, "Policy Design: Ubiquitous, Necessary and Difficult," *Handbook of Public Policy*, B. G. Peters and J. Pierre, eds., SAGE Publications Ltd, London, UK, pp. 75–96.

[33] Ouyang, X., 2021, "Policy Learning and Temporal Dynamics in Policy Mix: A Comparative Study of Green Energy Innovations," *Policy Sciences*, 54(2), pp. 305–328.

[34] Wu, X., Ramesh, M., and Howlett, M., 2015, "Policy Capacity: A Conceptual Framework for Understanding Policy Competences and Capabilities," *Policy and Society*, 34(3–4), pp. 165–171.

[35] Capano, G., and Woo, J. J., 2020, "Agility and Robustness as Design Criteria," *Routledge Handbook of Policy Design*, M. Howlett and I. Mukherjee, eds., Routledge, London, UK, pp. 420–434.

[36] Jordan, A., Wurzel, R. K. W., and Zito, A., 2013, "Still the Century of 'New' Environmental Policy Instruments? Exploring Patterns of Innovation and Continuity," *Environmental Politics*, 22(1), pp. 155–173. https://doi.org/10.1080/09644016.2013.755839

[37] Mistree, F., Hughes, O. F., and Bras, B., 1993, "Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm," *Progress in Astronautics and Aeronautics*, Washington, DC, USA, Vol. 150, pp. 251–251.

[38] Marston, M., Allen, J., and Mistree, F., 2000, "The Decision Support Problem Technique: Integrating Descriptive and Normative Frameworks," *Engineering Valuation and Cost Analysis*, 3(2), pp. 107–129.

[39] Guo, L., Allen, J. K., and Mistree, F., 2024, "Optimize or Satisfice in Engineering Design?" *Research in Engineering Design*, 35(3), pp. 239–267. https://doi.org/10.1007/s00163-023-00431-5

[40] Sharma, G., Allen, J. K., and Mistree, F., 2022, "Designing Concurrently and Hierarchically Coupled Systems," *Engineering Optimization*, 55(9), pp. 1556–1576.

[41] Mistree, F., Smith, W. F., Kamal, S. Z., and Bras, B. A., 1991, "Designing Decisions: Axioms, Models and Marine Applications," *Fourth International Marine Systems Design Conference*, Kobe, Japan, pp. 1–24.

[42] Nellippallil, A. B., Allen, J. K., Gautham, B. P., Singh, A. K., and Mistree, F., 2020, "Robust Concept Exploration of Materials, Products, and Associated Manufacturing Processes," *Architecting Robust Co-Design of Materials, Products, and Manufacturing Processes*, Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-030-45324-4_7

[43] Baby, M., Rama Sushil, R., Ramu, P., Allen, J. K., Mistree, F., and Nellippallil, A. B., 2024, "Robust, Co-Design Exploration of Multilevel Product, Material, and Manufacturing Process Systems," *Integrating Materials and Manufacturing Innovation*, 13(1), pp. 14–35.

[44] Sushil, R. R., Baby, M., Sharma, G., Nellippallil, A. B., and Ramu, P., 2025, "A Novel Visualization-Enabled Decision Support Framework for Data-Driven Integrated Design Space Exploration," *Structural and Multidisciplinary Optimization*, 68(1), pp. 1–20.

[45] Huey, C., Metghalchi, H., and Levendis, Y., 2024, "E-Fuels as Reduced Carbon Emission Options," *ASME Open Journal of Engineering*, 3, p. 031017.

[46] Jayaraman, R., Colapinto, C., La Torre, D., and Malik, T., 2017, "A Weighted Goal Programming Model for Planning Sustainable Development Applied to Gulf Cooperation Council Countries," *Applied Energy*, 185, pp. 1931–1939.

[47] Jia, R., Liu, Y., and Bai, X., 2020, "Distributionally Robust Goal Programming Approach for Planning a Sustainable Development Problem," *Journal of Cleaner Production*, 256, p. 120438.

[48] "World Bank Open Data," accessed March 2, 2025, https://data.worldbank.org

[49] "Transforming Our World: The 2030 Agenda for Sustainable Development," Department of Economic and Social Affairs, accessed March 2, 2025, https://sdgs.un.org/2030agenda

[50] "IEA-International Energy Agency," accessed March 2, 2025, https://www.iea.org/data-and-statistics

[51] Ritchie, H., Rosado, P., and Roser, M., 2023, "CO_2 and Greenhouse Gas Emissions," published online at OurWorldinData.org, accessed March 2, 2025, https://ourworldindata.org/co2-and-greenhouse-gas-emissions

CHAPTER 6

AUGMENTING DESIGN INTELLIGENCE: GENERATIVE AI AND KNOWLEDGE GRAPHS FOR INTERPRETABLE TRADE-OFF NAVIGATION

Background and Context

The design problems we address in Chapters 4 and 5 are tractable for a human designer working directly with the cDSP formulations and their outputs. Four goals, eight decision variables, thirty-three scenarios: a designer can visualize these, can hold the trade-off structure in working memory, can navigate the iSOM plots with genuine understanding. The satisfaction of that tractability should not be mistaken for a claim about the typical scale of E-CPSS design problems. Hot rod rolling involves six coupled objectives spanning two design levels with one hundred and sixty-nine scenarios. In supply chains, designers face dozens of interacting decision variables under non-stationary dynamics, and conditions change before they can act on a scenario analysis. Real agricultural systems involve not five goals but potentially twenty, spanning multiple regions with heterogeneous stakeholder populations. When the design problem reaches this dimensionality, the satisficing philosophy we established in Chapter 4 remains the right governing orientation, but the methods we have built so far place the full cognitive burden of navigation on the designer. That burden grows faster than any human decision process can accommodate.

The response cannot be to abandon interpretability in favor of automated optimization. A designer working in an E-CPSS context with real governance constraints, real stakeholder accountability, and real uncertainty about which scenarios will materialize must be able to explain and justify the design rationale. A solution produced by a black-box optimizer that the designer cannot interpret, justify, or interrogate is not a design solution in the Strategic Engineering sense; it is an answer in search of a question. What is needed is a method that expands the designer's effective cognitive reach without removing the designer's judgment from the process. This is what DSIM Layer 3, the Augmentation Layer, provides.

We develop a three-step framework that combines a Large Language Model with a Knowledge Graph to produce the three capabilities a designer needs simultaneously when navigating a high-dimensional satisficing space: interpretability in the form of narrative trade-off explanations that translate numerical scenario outcomes into human-legible design rationale, adaptivity in the form of targeted infeasibility recovery guidance when preference shifts or threshold tightening close off the accessible solution space, and a structured knowledge architecture that persists across the design session and can be queried semantically rather than exhausted sequentially. The LLM extracts candidate knowledge statements from structured scenario data, each statement is validated quantitatively against scenario outcomes before Knowledge Graph ingestion, and the validated graph supports interactive querying in a conversational interface. We demonstrate the framework on the hot rod rolling steel manufacturing test problem, one hundred and sixty-nine scenarios, six objectives across two coupled design levels, a dimensionality that makes conventional visualization approaches uninformative for decision-making.

The output of this chapter is DSIM Layer 3, the Augmentation Layer, and through it the validated Knowledge Graph that forms the K(si) component for the hot rod rolling context. We address RQ3a in full. The Knowledge Graph produced here is not only the contribution of Chapter 6; it is the foundation that the autonomous agents of Chapter 7 draw upon when they must execute strategies at the timescale of the system rather than the timescale of human deliberation. The next challenge the DSIM must address is speed. When E-CPSS dynamics unfold faster than any human design cycle allows, augmented cognition is still not enough. Execution must become autonomous, and the chapter that follows is where we build the architecture for that.

Preface

In multi-objective manufacturing designers face increasing difficulty in visualizing and interpreting trade-offs as the number of goals and decision variables grows. Traditional visualization methods offer intuitive insights for three-goal problems but do not scale effectively to higher dimensions. In this chapter we introduce a Generative Artificial Intelligence (GenAI) framework to support trade-off analysis in manufacturing decision-making which can further be generalized. We design the framework so that designers generate interpretable satisficing solutions and judge whether those solutions meet all key requirements at acceptable levels without optimizing one metric at the expense of the others. In the proposed framework we employ Large

Language Models (LLMs) to produce narrative explanations and adaptive interpretations of high-dimensional trade-offs, to help designers to explore feasible solution regions and understand conflicts among goals. Using the Hot Rod Rolling (HRR) steel manufacturing process chain problem as a test problem, we show how GenAI can synthesize the complex relationships among six goals and overcome the interpretability limitations of static methods which is further scalable to higher numbers of goals. By using LLM's we propose a structured method to summarize trade-offs and suggest balanced weight allocations to restore feasibility when strict constraints lead to an empty solution set. Additionally, in this chapter we incorporate an LLM-based knowledge graph implementation, enhanced from an open-source version to structure extracted insights and support scalable decision analysis. Cumulatively, in this chapter we provide a pathway for scalable, explainable, and human-centered decision support in complex manufacturing systems.

Keywords: Generative Artificial Intelligence (GenAI), Large Language Models (LLMs), Satisficing solutions, High-dimensional decision-making, Manufacturing system design, Knowledge extraction.

Research Question, Hypothesis, and New Knowledge in this Chapter

Research Question	RQ3a. What are the principles and method a designer uses to enable interpretable and adaptive trade-off navigation when E-CPSS design problems exceed human cognitive tractability?
Hypothesis	Hypothesis 3a. By deploying a generative AI framework in which a large language model extracts knowledge statements from structured scenario data, each statement is validated quantitatively before Knowledge Graph ingestion, and the validated graph supports interactive querying and adaptive infeasibility recovery, we navigate high-dimensional satisficing spaces with interpretable and actionable guidance for design decision-making.
New Knowledge	New Knowledge 3a. A GenAI-anchored Knowledge Graph framework for augmenting E-CPSS design decision-making, with LLM-based knowledge extraction, quantitative validation protocols, and interactive trade-off summarization for interpretable guidance in high-dimensional design spaces.

6.1 Frame of Reference

6.1.1 Background in Multi-Objective Design and Satisficing Decisions

Engineering design problems often involve multiple conflicting objectives, like maximizing performance, minimizing cost, and ensuring safety simultaneously. Classical approaches to multi-objective optimization yield a Pareto-optimal solution set representing trade-offs among objectives [1]. These Pareto-optimal solutions are those for which no objective can be improved without sacrificing another, forming a Pareto front that the designer must navigate. In practice, selecting a single "optimal" design from this front is non-trivial, as multiple stakeholder preferences and uncertainty factors come into play. Rather than purely optimizing, designers frequently adopt satisficing strategies, aiming for solutions that *satisfy* all key requirements to an acceptable level rather than optimizing one metric at the expense of others. This concept of satisficing, rooted in Herbert Simon's work on bounded rationality [2], suggests that decision-makers settle for "good enough" solutions under real-world cognitive and information constraints. In engineering design, satisficing aligns with seeking a *ranged set* of solutions that meet all critical thresholds, instead of a single extreme optimum[3].

Over the past several decades, design researchers have formalized satisficing decision-making through constructs like the Compromise Decision Support Problem (cDSP) [4]. The cDSP formulation provides a mathematical means to model engineering decisions involving multiple goals and constraints, enabling designers to find solutions that *simultaneously satisfy* all goals within acceptable ranges [3]. Notably, Mistree and co-authors in the 1990s championed the cDSP as a way to generate "good enough" designs that balance competing objectives [3,4,5]. A solution to a cDSP is defined not by a single optimum value for each objective, but by a point in the design space that meets all goals satisfactorily; in other words, a design that "suffices" on all requirements and thereby *satisfices* the decision criteria [3]. We contrast our approach with traditional optimization and we emphasize feasible robustness over extreme optimality. We extended the framework and we introduced robust satisficing ideas for systems that must withstand uncertainties in manufacturing and usage. For example, Chen and co-authors [5] integrated Taguchi's robust design philosophy with multi-objective cDSP formulations, allowing designers to minimize

sensitivity to noise factors while meeting performance targets [3]. Robust decision support formulations, such as the robust coupled cDSP (rc-cDSP), explicitly account for variations in variables and model uncertainty, seeking solutions that remain feasible and near-optimal under a range of scenarios (i.e., *solution insensitivity* to uncertainty) [5]. Recent applications in complex systems design use these constructs to explore a ranged set of satisficing solutions that achieve multi-level goals with minimal performance degradation under perturbations.

Despite these advances, supporting human decision-makers in navigating multi-objective trade-offs remains challenging. It is well recognized that presenting a designer with an unstructured Pareto set can be overwhelming, especially as the number of objectives grows beyond two or three. Visualization and decision support tools are essential to make the trade-offs understandable. Traditional techniques (e.g., parallel coordinate plots, scatter plot matrices) can illustrate low-dimensional Pareto fronts, but visualizing high-dimensional objective spaces is non-trivial [6]. To aid intuitive decision-making, researchers have developed methods like self-organizing maps that project high-dimensional Pareto solution sets onto interpretable 2D plots. For instance, Nagar and co-authors [7] proposed an interpretable Self-Organizing Map (iSOM) approach to visualize many-objective Pareto fronts [6]. We use this method to produce a series of component planes that show each objective's structure and that reveal patterns even when we analyze more than three objectives [7]. In the follow-up work it was demonstrated that such visualization-aided decision-making can meaningfully improve designers' ability to select solutions in multi-criteria problems [6]. These human-in-the-loop tools reflect a broader emphasis on human-centered design in engineering decision support – solutions must not only exist mathematically, but also be *interpretable and explorable* by humans with domain expertise.

6.1.2 Gaps in Scalability, Interpretability, and Adaptivity of Current Approaches

While the multi-objective decision-support methodologies discussed above have proven useful on various design problems, several critical gaps and limitations have become apparent. One major concern is scalability. Designers meet the curse of dimensionality both when they explore the solution space and when they visualize trade-offs. Designers using current robust satisficing frameworks cannot handle high-dimensional design spaces efficiently, particularly in multilevel systems [9]. For example, in modern manufacturing system design, decisions span product,

material, and process levels, each with their own objectives and uncertainties[8]. Existing robust co-design methods (e.g., rc-cDSP-based frameworks) allow formulating such problems, but simultaneously exploring all levels' design spaces under uncertainty is computationally intensive [8]. Baby and co-authors [9] note that exploring coupled high-dimensional spaces can lead to combinatorial explosion, and even with machine-learning surrogates the search may become intractable or exceedingly slow . Thus, scalability in both problem size and number of objectives remains a bottleneck. New approaches are needed to accelerate design space exploration without sacrificing solution quality or diversity – for instance, by intelligently sampling the space or leveraging transfer learning across levels.

A second gap lies in interpretability and user insight. Many multi-objective optimization algorithms (genetic algorithms, Pareto solvers, etc.) function as black boxes, outputting a set of nondominated solutions without explanations. This can leave human designers struggling to discern *why* certain designs are better or how to navigate the trade-offs. As a result, there is growing emphasis on making the decision-support process more transparent and explainable. Researchers highlight in recent reviews that lack of interpretability is a significant concern with advanced design optimization and AI techniques [10, 11]. Designers who work with AI-generated solutions often cannot understand the model's reasoning or the relationships between design variables and objectives, and this hinders their trust and adoption of such tools [12]. For instance, a complex neural-network-based surrogate might predict performance accurately but not reveal which design features drive those predictions. Similarly, an evolutionary algorithm might return a Pareto front without indicating the *trade-off rationale* between two solutions[13]. This opacity is problematic in engineering domains where safety and rationale are paramount. The field of explainable AI (XAI) has begun to influence design optimization, spawning techniques to generate human-interpretable explanations for algorithmic decisions. However, most XAI in engineering to date has focused on explaining *input-feature importance* to explain which design parameters most affect an objective and has neglected the explanation of outputs and objectives trade-offs [11,14,15]. In multi-objective design, decision-makers need insight into *why one solution is preferable over another in terms of objectives* – for example, *"Solution A meets safety and cost goals better, whereas Solution B excels in performance"*. Without such narrative explanations, the onus is on the human to manually analyze data or run additional simulations to understand trade-offs, which is inefficient and error-prone. Clearly, more interpretable and human-centered

decision-support frameworks are needed so that engineers can *trust and effectively use* AI-generated design recommendations.

A third critical gap is adaptivity; the ability of design decision methods to dynamically adjust to new information, evolving preferences, or changing operational scenarios [16]. Traditional design optimization is typically a static, one-shot process: problem parameters (objectives, constraints, weights) are set *a priori*, an optimization run produces a fixed Pareto set, and then the process ends. In reality, engineering design is often an iterative, dynamic decision process. Requirements can change, new stakeholder preferences can emerge, or external factors (e.g., supply chain disruptions, new regulations) can alter the decision context. Current multi-objective frameworks offer limited support for this kind of adaptivity. If a decision-maker wishes to explore a "what-if" scenario or adjust preferences on the fly, usually a completely new optimization run or extensive sensitivity analysis is required, which is time-consuming[17,18]. There is a need for decision-support systems that can incorporate human feedback and evolve the solution space exploration in real-time. Some interactive methods in the literature (e.g., interactive evolutionary algorithms, preference-based optimization) address this by iteratively querying the user for preferences and updating the search [19]. However, even these can become cumbersome as the number of iterations or scenarios grows. In robust design under uncertainty, adaptivity also means the ability to handle multiple future scenarios efficiently [20]. Scenario-based robust optimization often produces a set of solutions that hedge against various possible futures but presenting and updating these as scenarios change is challenging. Techniques like scenario clustering have been proposed to reduce complexity by grouping similar uncertainty scenarios [21], yet integrating such approaches seamlessly into the design process is still an open problem. In summary, today's tools lack the fluid, *conversational* adaptivity that would allow a human designer to say, "Show me solutions favoring sustainability more heavily," or "How does the solution change if material costs double?" and immediately get updated guidance.

These gaps like scalability, interpretability, adaptivity, limit the effectiveness of current multi-objective and robust design methodologies in complex manufacturing system design. The consequence is that designers may not fully explore the design space (missing innovative solutions), may mistrust or misinterpret the solutions they do obtain, and may struggle to apply the results in a fast-changing context. There is a clear opportunity for new frameworks that address these issues, thereby enabling more efficient, insightful, and responsive decision-making in

engineering design.

6.1.3 Opportunities at the Intersection of Generative AI and Design Decision-Making

Addressing the above challenges calls for a paradigm shift in how we support design exploration and decisions. Generative Artificial Intelligence (GenAI) encompassing technologies like large language models (LLMs), generative adversarial networks (GANs), variational autoencoders (VAEs), diffusion models, and vision-language models offers a promising toolkit to enhance scalability, interpretability, and adaptivity in design. In the past few years, generative AI has undergone explosive advancement and adoption. This trend reflects the excitement around tools such as ChatGPT (a text-generating LLM) and DALL·E 2 / Stable Diffusion (image-generating models), which have showcased the power of AI to produce creative outputs from simple prompts. However, applying generative AI in engineering design requires going beyond surface-level generation of designs to embed engineering knowledge, constraints, and context into the AI's generative process [14,17]. Recent studies caution that off-the-shelf generative models tend to "mimic" existing designs rather than truly innovate, often failing to improve on engineering performance metrics because the models are optimized only for similarity to training data [17]. We argue that the knowledge creation and exploration stage is a critical aspect in this regard. To unlock GenAI's potential in design, researchers argue that the models' objectives must be refocused towards *design requirements and functional performance*, not just data reconstruction [11]. In other words, generative models need to be guided to explore explainable, high-performing regions of the design space, effectively combining AI's creativity with rigorous engineering evaluation.

The proposed work positions itself precisely at the intersection of generative AI and robust, interpretable decision-making. By leveraging generative AI in a *human-centered* framework, we aim to tackle the earlier identified gaps as follows:

- Scalability: Generative models are great at learning complex high-dimensional distributions from data, which can be harnessed to improve design space exploration. Techniques like VAEs and diffusion models can learn latent representations of feasible design solutions, enabling rapid generation of new candidates that probabilistically satisfy constraints. In a multi-objective context, one could train a generative model (for example, a conditional GAN or

diffusion net) on known Pareto-optimal or near-optimal solutions and then use it to *sample the solution space efficiently*. We accelerate the search for satisficing solutions with this approach by focusing on high quality regions that we identify from data. Moreover, generative surrogates can approximate expensive simulations, providing nearly instantaneous predictions of performance for generated designs. Prior work has shown that incorporating advanced generative methods (GANs, etc.) alongside traditional search techniques yields orders-of-magnitude more design alternatives without prohibitive cost [22, 23]. We foresee a GenAI-enabled explorer that can quickly suggest a diverse set of designs satisfying multiple goals, effectively scaling problems with many objectives or deeply coupled variables. The generative model acts as a smart guide through the design space, flagging promising regions and skipping implausible ones, thereby addressing scalability both computationally and cognitively.

- Interpretability: Generative AI can also enhance interpretability by providing natural-language and visual explanations of design trade-offs. A novel idea is to integrate LLMs (like GPT-4) as an explanation engine attached to the technical optimization. For example, once a set of Pareto-optimal solutions is generated, an LLM could be prompted to produce a narrative summary of the trade-offs – e.g., *"Design A offers the best strength and moderate cost, whereas Design B sacrifices some strength for a 20% cost reduction. Design C balances the two, providing a middle-ground solution."* Such narrative explanations, generated from the quantitative data, would make the Pareto front more approachable to human decision-makers. Recent research on explainable AI for decision-making supports this vision: Chakraborty and coauthors (2025) introduced MOLONE (Multi-Output LOcal Narrative Explanation) to explain preference selections in multi-objective optimization by highlighting the importance of each objective in comparing two alternatives [14]. Their user studies confirmed that human-in-the-loop decisions improve significantly when provided with comparative narrative explanations, essentially because the decision-makers better understand the *why* behind the AI's suggestions [14]. Building on this insight, our framework will use generative narrative techniques to accompany every design recommendation with human-readable justification. In addition, vision-language models (VLMs) could be employed to generate annotated visualizations of trade-offs (for instance, automatically labeling regions of a parallel coordinates plot with plain-language insights). We address the interpretability gap: rather than the designer deciphering a cloud of Pareto points, the AI will present explanations and even analogies (leveraging its

training on vast text corpora) to contextualize the options. Ensuring that decisions remain human-centered while AI provides explanatory support will be a core guiding principle of the framework.

- Adaptivity: Generative AI models are inherently adaptive in that they can be conditioned on new inputs and updated with additional training data on the fly. We will exploit this by creating an interactive loop between the human designer and the AI. Imagine a system where the designer can input a change (a new preference, or a new scenario) in natural language – e.g., *"What if material costs increase by 50%?"* – and the generative model adapts the solution space exploration accordingly in real time. Large language models can parse such user queries and modify the problem formulation or constraints that are fed into the optimization/generative modules. On the backend, one could maintain a Bayesian preference model (such as in Preference-Based Optimization) that updates as the user provides feedback on generated solutions, thus steering the generative model towards the user's evolving preferences. The narrative capabilities of the LLM can also help cluster and summarize scenario-based solution sets. For example, if exploring robust solutions under 100 possible scenarios of demand and supply uncertainty, the AI could cluster these scenarios (using unsupervised learning on scenario features) into a few archetypes and provide a summary: *"Scenarios fall into three groups – low-demand, medium-demand, high-demand – and here is the best satisficing design for each group."* This kind of dynamic scenario clustering and explanation would greatly aid the designer in understanding a complex robust solution space. Early efforts in robust optimization have used scenario clustering algorithms to reduce complexity [21], and here we propose to take it further by integrating it with *semantic, explainable outputs* from an AI assistant. The overall vision is a conversational, adaptive decision-support framework: the generative AI produces candidate solutions and explanations, the human provides feedback or new questions, and the AI dynamically responds with refined solutions or additional insight. Such a system continuously adapts to the decision-maker's needs and the problem's context in real time, a capability far beyond the static processes common today.

In summary, the convergence of generative AI and design decision-making promises to overcome current limitations by marrying computational power with human-centered insight. Generative models can drastically improve the breadth and depth of design exploration (enhancing scalability), while their ability to produce language and visual content can turn raw data into

meaningful stories and interactions (enhancing interpretability and adaptivity). This aligns perfectly with the needs identified in complex manufacturing system design, where one must explore vast solution spaces, comprehend intricate trade-offs, and respond swiftly to new information. Recent authoritative reviews have called for exactly this integration – noting opportunities to bring diffusion models and LLMs into the engineering design toolkit, and to treat AI as a co-designer that augments human creativity and decision-making [20].

6.1.4 Positioning of the Proposed Work

We situate this work at the intersection of generative AI, multi-objective robust design, and human-centered decision support. We contribute to the emerging discourse on 'Generative AI in Design and Manufacturing' by demonstrating the utility of GenAI and knowledge graphs not merely for automating design generation, but for empowering human designers with enhanced tools to explore alternatives and make informed decisions. We build upon foundational concepts like the cDSP and the rc-cDSP for structured decision problem modeling [2, 4], while injecting new capabilities through AI-driven modules for generation, explanation, and interaction. By emphasizing interpretable satisficing (rather than opaque optimizing), we address the gap in trust and transparency, a key requirement for adoption in industry. Additionally, by focusing on complex manufacturing design scenarios (characterized by multi-level decisions, uncertainty, and large data), we ensure the developed methods are scalable and generalizable to industrial operational scenarios that matter.

In doing so, we respond to identified research gaps and recent calls to action in the literature. It provides a pathway to achieve scalable co-design exploration (through generative sampling and learning) as implied by recent multilevel design studies [8], and answers the need for narrative, dynamic decision support as evidenced by the success of methods like MOLONE in human-in-the-loop optimization [14]. Ultimately, the outcome will be a human centered framework and prototype system in which a generative AI acts as a *"co-pilot"* for the engineer – dynamically generating design alternatives, explaining the trade-offs in plain language, and adapting to the designers inputs. This represents a significant step toward AI-assisted design exploration, where instead of replacing human designers, the AI augments their capabilities to navigate complexity, much like a GPS guides a traveler through a labyrinth of possibilities.

By situating the work at this transdisciplinary intersection, we aim to advance knowledge in

multiple domains: (1) *Design Principles* ; by introducing a new decision-support paradigm blending formal decision models with AI-driven exploration; (2) *Artificial Intelligence in Engineering*; by extending the application of generative models and AI techniques to engineering design problems, demonstrating how to incorporate domain-specific constraints into foundation models by augmenting the utility of knowledge graphs; and (3) *Human-Computer Interaction for Engineering Systems*; by developing interactive mechanisms for conversationally steering optimization processes. The insights gained will not only fill current gaps (scalability, interpretability, adaptivity) but also lay the groundwork for future design environments where engineers and generative AI collaborate seamlessly to solve complex design challenges. We present this work as an early example of how generative AI can support human-centered multi-objective design, with the aim of encouraging further research and development in this area.

6.2 Proposed Framework

We design this framework so that designers produce interpretable satisficing solutions in high-dimensional manufacturing problems. We structure it as a workflow in which designers move from raw scenario data to design decisions. The process unfolds in three major steps: *knowledge creation*, *knowledge exploration (via querying and clustering),* and *adaptive solution-space generation*. Together, with these steps, designers move seamlessly from data to interpretation and decision-making. In Figure 6.1 we illustrate the proposed framework as a multi-layered connection between the human designer, the AI decision-support engine, and the underlying data foundation. We depict the designer at the top layer in Figure 6.1. We invite the designer to prompt queries and to receive explanations and preferences that guide decision making. We represent the generative AI engine in the middle layer of Figure 6.1 and we use it to support *interactive querying*, *scenario clustering*, and *adaptive reasoning*. We provide the data foundation at the bottom layer and we curate scenario repositories that we map into knowledge representations such as knowledge graphs or databases. The layers are interconnected, enabling iterative feedback between the designer and AI engine, as well as continuous exchange between AI and the underlying data.

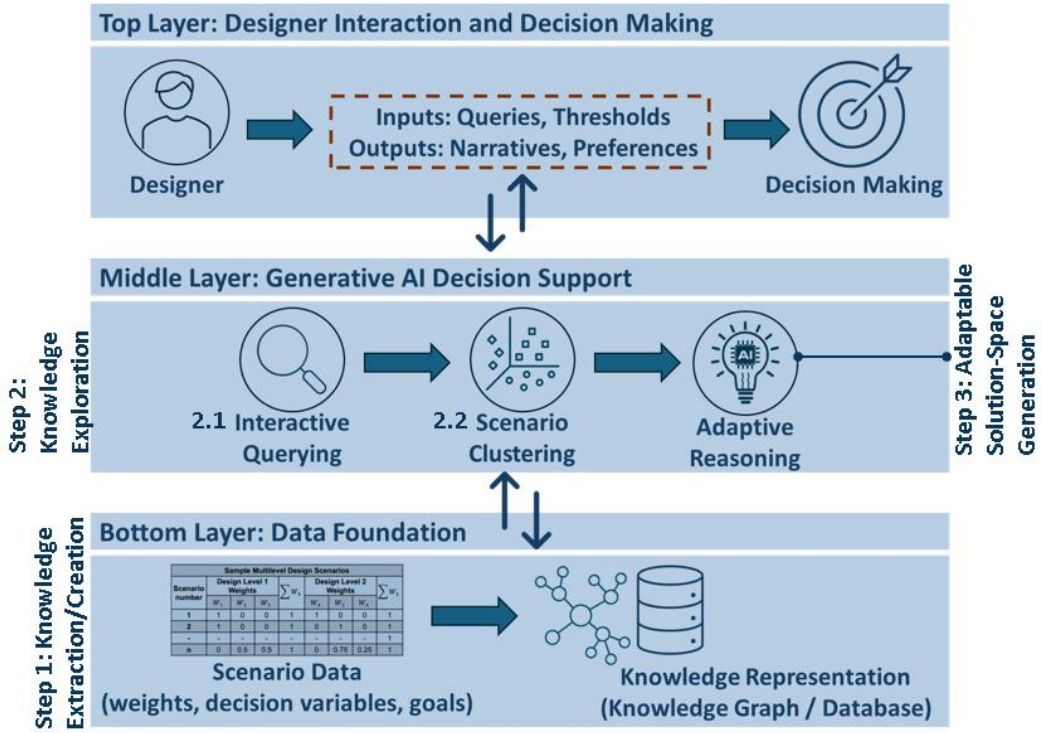


Figure 6.1: Proposed Framework Architecture to Support Interpretable Satisficing Solution Design in High Dimensional Spaces

Having outlined the overall structure of the framework in Figure 6.1, we now describe its operation step by step. The pipeline unfolds in three stages: *knowledge creation*, where scenario data are transformed into structured knowledge; *knowledge exploration*, where this knowledge is interrogated and clustered to reveal trade-offs; and *adaptable solution-space generation*, where feasible solutions are dynamically updated as designer preferences evolve.

Step 1: Knowledge Creation/Extraction

The foundation of the framework is the systematic analysis of scenario data, which consists of weight allocations, decision variables, and resulting goal values. By extracting correlations, identifying conflicts, and mapping dependencies between goals and decision variables, Through the framework we transform tabular data into a structured representation of knowledge. To operationalize this, each of the 169 rc-cDSP scenarios is first converted into a structured JSON schema that explicitly encodes Level-1 and Level-2 weights, decision variables, and the resulting performance goals. This structured representation functions as the semantic data used by the LLM. The JSON-encoded scenarios are then supplied to the LLM using a few-shot prompting strategy.

Several example JSON–interpretation pairs are provided to the model to establish a consistent pattern for extracting relationships such as synergies, conflicts, sensitivities, and directional influences. This prompting strategy enables the LLM to generate interpretable knowledge statements directly from the structured data without any parameter updates or model training.

While large language models can generate coherent natural-language summaries of design trade-offs, directly converting such statements into a consistent and machine-readable knowledge graph requires strict adherence to a predefined schema and relational vocabulary. General-purpose prompting alone does not reliably guarantee this level of structural consistency, particularly when the output must conform to a fixed set of node types and relationship types. To address this gap, we introduce a lightweight fine-tuning step that enables deterministic transformation of validated natural-language knowledge statements into structured subject–relation–object triples. This step serves purely as a structural post-processing mechanism and does not perform reasoning, discovery, or inference of new engineering relationships. In the proposed framework, foundation models are therefore used for knowledge extraction via prompting, while a separately fine-tuned model is used only to enforce schema-constrained representation within the knowledge graph. Through these structured representations, we allow subsequent AI modules to reason semantically over the data and provide interpretable insights, moving beyond traditional static visualizations and interpretations. Therefore, we employ an LLM-based knowledge graph as an illustrative tool that connects goals, weights, and outcomes in a more intuitive and interpretable form.

In the proposed framework, knowledge extraction includes an internal quantitative verification step: candidate statements are generated from the scenario data and are retained only if they are supported by dataset-derived metrics appropriate to the type of claim. The specific metrics used for this verification may vary by application; in the HRR test problem, extracted statements are assessed using correlation analysis, sensitivity measures, and trade-off quantification. Only statements that satisfy these criteria are reported as extracted knowledge and passed to subsequent stages, with implementation details provided in Section 6.4.

Step 2: Knowledge Exploration

Once the knowledge base is established, designers require tools to explore and interpret the solution space. This exploration occurs along two parallel tracks:

- ***Step 2.1 Interactive Querying***

Designers can directly interrogate the database or the knowledge graph using natural-language queries supported by large language models. Instead of manually scanning hundreds of rows of data, the LLM can filter and synthesize information into concise responses. For example, queries such *as "Which scenarios maximize Goal 5 without reducing Goal 3 below its baseline?"* or *"What can I change to have more robust values for Goals 1–3?"* are answered by identifying candidate scenarios, presenting trade-offs, and highlighting relevant decision-variable patterns. Querying through the knowledge graph further enriches this process by enabling semantic reasoning over the structured relationships between goals, weights, and outcomes. This interaction ensures that knowledge is not only accessible but also actionable in real time. While iSOM provides interpretable component-plane visualizations, acting on infeasibility typically requires designers to manually inspect multiple planes, infer the dominant conflicts, adjust thresholds, and re-run the visualization iteratively. In contrast, through our framework we reduce this manual burden by returning (i) feasibility status, (ii) an explicit explanation of the primary conflict drivers, and (iii) targeted threshold relaxations to restore feasibility in a single conversational step, enabling faster "what-if" iteration without exhaustive plot scanning

- ***Step 2.2 Scenario Clustering and Narratives***

In parallel with querying, we apply embedding-based clustering to group scenarios with similar trade-off behaviors. Each cluster is then interpreted and described in human terms, such as *"Cluster A: energy-efficient but low strength"* or *"Cluster B: balanced strength and cost but high emissions."* The clustering step provides a high-level view of the solution landscape, complementing query-based exploration. It reveals design archetypes that optimization alone may fail to expose, allowing designers to compare categories of solutions rather than isolated scenarios.

Step 3: Adaptable Solution-Space Generation

In the final step we focus on adaptation. In this section, "adaptivity" refers to adaptive human-in-the-loop exploration of an existing rc-cDSP solution space, such as iteratively updating feasibility and recommendations through preference/threshold adjustment over the pre-computed 169-scenario library, rather than generating new rc-cDSP scenarios. It is common for designers to adjust their preferences as priorities, constraints, and uncertainties evolve during the design process. As designers refine their preferences or update thresholds for satisficing goals, the knowledge base is re-queried, and feasible solution sets are dynamically updated. For example,

tightening a constraint on Goal 1 may reduce the set of acceptable scenarios, whereas relaxing the same constraint can broaden the solution space and reveal new trade-offs with Goals 3 or 5. By iteratively updating the solution space, we ensures that the decision process remains flexible and responsive to evolving requirements. The adaptive capability not only reflects the dynamic nature of real-world manufacturing but also ensures that designers maintain control while navigating complex trade-offs.

Human-Centered Decision Support

Through the proposed framework we incorporate the human in the loop by embedding principles of human-centered decision support into the exploration of multi-objective solutions. Rather than replacing human judgment, we enhance the designer's agency by providing interpretable narratives, actionable insights, and tools for guided exploration of the solution space. Through clustering, interactive querying, and trade-off summarization, we provide designers with a structured process for translating complex numerical outputs into accessible explanations that align with their reasoning in decision-making. We ensure that human decision-makers remain central, while also offering scalability to higher-dimensional design spaces where traditional visualization and analysis techniques fall short.

6.3 Test Problem: Hot Rod Rolling (HRR)

Product–Material–Manufacturing Process System Design

Hod Rod Rolling Problem To demonstrate our proposed framework, we adopt the Hot Rod Rolling (HRR) problem as a test problem. The HRR system is an industry-inspired manufacturing process chain used to produce hot-rolled steel rods. It begins with reheating steel billets, which are then plastically deformed into rods through a sequence of rolling mills, followed by controlled cooling on a run-out table. This thermo-mechanical sequence drives microstructural evolution, such as changes in ferrite grain size, pearlite spacing, and ferrite fraction, which in turn determine the rods' mechanical properties including yield strength, tensile strength, and hardness. To meet performance targets, design decisions must integrate processing conditions, material microstructure, and final product properties. In the referenced study, we scope the HRR Product–Material–Manufacturing Process (PMMP) to focus on two design levels (i) material design

variables and (ii) the cooling process to demonstrate how co-design frameworks can capture interdependencies and manage trade-offs across levels.

In the original study [8], the Interpretable Self-Organizing Map (iSOM) was employed as the visualization tool for design space exploration. To highlight the capabilities of our proposed generative-AI framework, we apply it to the same problem and contrast its performance with the iSOM-based approach. iSOM provides an established, human-interpretable way to explore trade-offs in the Hot Rod Rolling (HRR) problem and therefore serves as a useful reference point for evaluating whether our generative-AI–based interpretation is equally reliable, or provides complementary insights. For the test problem we use an industry-inspired Product–Material–Manufacturing-Process (PMMP). HRR is an ideal testbed because it couples processing, microstructure and product properties, produces multiple conflicting objectives, and yields a high-dimensional design space that is difficult to interpret using classical methods. The focus though the HRR problem is on co-design interactions between the materials (upper) level and the cooling process (lower) level. It is modeled using the robust, coupled compromise Decision Support Problem (rc-cDSP) constructed in the source work [8]. HRR converts reheated steel billets into hot-rolled rods via thermo-mechanical processing (reheat → rolling → controlled cooling). The thermo-mechanical history during rolling and cooling governs the final microstructure, ferrite grain size, pearlite spacing, ferrite fraction, which in turn determines mechanical properties such as yield strength, tensile strength, and hardness. Consequently, design choices at both the materials level and the cooling/process level are tightly coupled, creating trade-offs that must be managed jointly.

In Tables 1 and 2 we present the decision variables and performance goals for Design Level 1 (materials designer), while in Tables 3 and 4 we summarize the decision variables and performance goals for Design Level 2 (cooling process designer). All variables, goals, bounds, and targets are taken directly from the original study [8].

Table 6.1: Decision Variables of Design Level 1 (Materials designer)

Variable	Description	Bounds & Uncertainty	Unit
Xf	Ferrite fraction	0.1–1.0 ±0.1	-
Dα	Ferrite grain size	5–25 ±3	μm

S0	Pearlite interlamellar spacing	0.15–0.25 ±0.01	μm
[Mn]	Manganese concentration (shared variable)	0.7–1.5 ±0.1	%

Table 6.2: Performance goals of Design Level 1 (Materials designer)

Goal	Description	Target	Type
G1 - EMI YS	EMI of Yield Strength	3	Maximize
G2 - DCI TS	Design Capability Index for Tensile Strength	8	Maximize
G3 - DCI HV	Design Capability Index for Hardness	8	Maximize

Table 6.3: Decision variables of Design Level 2 (Cooling process designer)

Variable	Description	Bounds & Uncertainty	Unit
dγ	Austenite grain size (input microstructure)	30–100 ±10	μm
CR	Cooling rate	0.18–1.66 ±0.17	°C/s
[Mn]	Copy of [Mn] (consistency constraint)	0.7–1.5 ±0.1	%

Table 6.4: Performance goals of Design Level 2 (Cooling process designer)

Goal	Description	Target	Type
G4 - DCI Dα	DCI for ferrite grain size	3	Maximize
G5 - DCI S0	DCI for pearlite interlamellar spacing	10	Maximize
G6 - DCI Xf	DCI for ferrite fraction	10	Maximize

To provide a dataset for our generative AI analysis, we draw upon the method presented in the source paper. The HRR problem is modeled using rc-cDSP, which formally captures the decisions and interactions between the different design levels while accounting for uncertainties. Using this

model, a comprehensive set of design scenarios is generated. Baby and co-authors examined 13 distinct design scenarios at each of the two individual levels (material and process). By considering all possible combinations, a total of 169 multilevel design scenarios were created. Each scenario represents a unique combination of designer preferences (weights) for the different goals, resulting in a corresponding set of decision variable values and performance outcomes.

This dataset of 169 scenarios, defined by six goals and multilevel complexity, is too high-dimensional for straightforward human interpretation. It therefore serves as an ideal test problem for demonstrating how our generative AI framework can surpass traditional methods in summarizing trade-offs, enabling interactive exploration, and guiding designers toward robust, satisficing solutions. For the generative-AI analysis, we employed GPT-5 (OpenAI) to process the 169 scenarios, generate narrative explanations, and produce adaptive interpretations, as outlined in this chapter.

6.4 Results and Discussions

In this section, we discuss the results obtained by applying the proposed framework across its various steps.

6.4.1 Knowledge Extraction from Data (Step 1)

For the test problem, we input the 169 scenarios generated from the rc-cDSP construct [8] into the large language model. Instead of manually examining each scenario, the LLM synthesized the dataset into a series of interpretable knowledge statements that capture relationships, conflicts, and synergies among goals and decision variables. As described in Section 6.2, each scenario was first converted into a structured JSON schema that explicitly encodes Level-1 and Level-2 weights, decision variables, and resulting goal values. This structured representation serves as the semantic data foundation that enables the LLM to interpret design interactions in a consistent format. An example JSON representation for Scenario 12 and 40 used in the HRR study is shown in Table 6.5.

To guide the LLM in generating interpretable statements, we employed a few-shot prompt engineering strategy in which several example JSON-interpretation pairs were provided. This prompting protocol teaches the model the expected structure of the extracted knowledge without requiring any model training or fine-tuning. A representative excerpt from the few-shot prompt template is shown in Table 6.5.

Table 6.5: Structured JSON Scenario Examples and Few-Shot Prompt Used for Knowledge Extraction

[System Instruction: You are an analytical assistant that interprets multi-objective design scenarios.
Given a structured JSON description of a scenario, extract concise statements describing relationships, synergies, conflicts, sensitivities, or threshold effects among goals, weights, and decision variables.
Your output must:
- contain 1–2 declarative statements,
- refer only to patterns supported by the numerical values in the JSON,
- avoid speculation or invented trends,
- highlight trade-offs across design levels when relevant.

Few-Shot Example 1
Input (JSON):

```
{
  "scenario_id": 12,
  "level_1": {
    "weights": {"YS": 0.18, "TS": 0.47, "HV": 0.35},
    "goals":  {"EMI_YS": 0.98, "DCI_TS": 6.72, "DCI_HV": 7.88}
  },
  "level_2": {
    "weights": {"Dα": 0.22, "S0": 0.51, "Xf": 0.27},
    "goals":  {"DCI_Dα": 1.65, "DCI_S0": 9.82, "DCI_Xf": 10.44}
  }
}
```

Output (Knowledge Statements):
- Lower YS weight is associated with EMI_YS falling below 1.0.
- Emphasis on S0 at Level 2 improves DCI_S0 while slightly reducing DCI_Dα.

Few-Shot Example 2

Input (JSON):

```
{
  "scenario_id": 40,
  "level_1": {
    "weights": {"YS": 0.52, "TS": 0.28, "HV": 0.20},
    "goals":  {"EMI_YS": 1.45, "DCI_TS": 3.25, "DCI_HV": 4.62}
  },
  "level_2": {
    "weights": {"Dα": 0.35, "S0": 0.42, "Xf": 0.23},
    "goals":  {"DCI_Dα": 2.00, "DCI_S0": 10.40, "DCI_Xf": 10.21}
  }
}
```

Output (Knowledge Statements):

- High YS weight increases EMI_YS but reduces DCI_TS and DCI_HV.

- Xf remains robust even with moderate weight, suggesting low sensitivity.

New Scenario (JSON):

{scenario_data_here}

Task:

Based on the patterns demonstrated in the examples above, extract 1–2 concise knowledge statements describing the relationships among goals, weights, and decision variables in the new scenario. Focus on synergies, conflicts, sensitivities, threshold behaviors, and cross-level interactions.]

For the HRR test problem, we implement the validation step using the complete set of 169 rc-cDSP scenarios. Each extracted knowledge statement is evaluated using a quantitative test aligned with the type of claim it expresses. Relationship claims (such as “strong inverse” or “strong synergy”) are validated using pairwise Pearson correlation coefficients computed across all scenarios. Sensitivity and variability claims are validated through range and spread comparisons

of achieved performance metrics. Statements that describe explicit trade-off magnitudes are validated by comparing paired scenarios and computing relative percentage changes.

In this framework, LLM-generated statements are treated as hypotheses and are retained only if they pass the quantitative validation tests; unsupported statements are discarded and not propagated to the knowledge graph. In Table 6.6, we provide representative examples of how we quantitatively validate the knowledge statements that we extract. The first four examples illustrate supported statements retained in the final knowledge layer, while the last one show candidate statement that was excluded due to weak or inconsistent empirical support. Based on these tests, each statement is classified as supported, partially supported, or not supported which only supported statements are retained and reported. The quantitative evidence used for validation (correlation coefficients, ranges, and percentage changes) is preserved and later associated with the corresponding relationships in the knowledge graph.

Table 6.6: Sample of Quantitative Validation of Extracted Knowledge Statements

Statement (summary)	Claim type	Validation test	Quantitative evidence	Verdict	KG attribute
Strong inverse relationship between EMIYS and DCITS	Outcome–outcome relationship	Pearson correlation	r = −0.956	Supported	{correlation: −0.956}
Strong synergy between DCITS and DCIHV	Outcome–outcome relationship	Pearson correlation	r = +0.987	Supported	{correlation: +0.987}

Ferrite Fraction (DCI FF) shows highest sensitivity	Range / variability	Range comparison	Range(DCI FF) = 4.49, among highest across metrics	Supported	{range: 4.49}
Lowest EMI YS (<1.0) occurs when w_YS < 0.20	Threshold / conditional	Conditional frequency	61.5% of cases with w_YS < 0.20 have EMI YS < 1.0	Supported	{threshold: 0.20, frequency: 0.615}
DCI HV is strongly correlated with EMI YS	Outcome–outcome relationship	Pearson correlation	r = −0.971	Not supported	—

Using the structured input, few-shot prompting and validating extracted knowledge statements, LLM produced a set of 19 high-level knowledge statements summarizing global trade-off patterns across the 169 scenarios. In the following Table 6.7, we illustrate the raw knowledge extracted from the scenario data.

Table 6.7: Set of Knowledge Extracted by the LLM from the Curated HRR Scenario Dataset

1. There is a strong inverse relationship between EMI YS (Yield Strength) and DCI TS (Tensile Strength).
2. The goals for DCI TS (Tensile Strength) and DCI HV (Hardness) are strongly synergistic; increasing the weight on one almost always improves the other by a similar magnitude.

3. Prioritizing the Yield Strength goal is the only reliable way to achieve an EMI YS greater than 1.3, but this strategy results in the lowest performance for both DCI TS and DCI HV.

4. The robustness of Ferrite Fraction (DCI Xf) is highly sensitive to designer preferences, showing the widest performance range in the dataset (from ~8.5 to over 13.0).

5. Maximizing the weight on the Ferrite Fraction (Xf) goal severely penalizes DCI Dα (Ferrite Grain Size) and DCI S0 (Pearlite Interlamellar Spacing), causing their values to drop by 30–50%.

6. A high weight on the Pearlite Interlamellar Spacing (S0) goal complements the Ferrite Grain Size (Dα) goal, effectively boosting both robustness metrics simultaneously.

7. Achieving a DCI Dα value above 2.0 is challenging and typically requires a high weight on either the Ferrite Grain Size (Dα) or Pearlite Interlamellar Spacing (S0) goals at Level 2.

8. Holding Level 1 preferences constant, prioritizing a high Ferrite Fraction (Xf) at Level 2 leads to a marginal increase in EMI YS but a slight decrease in DCI TS.

9. The EMI YS value is primarily dictated by Level 1 weights; changes in Level 2 preferences only cause minor fluctuations.

10. The lowest EMI YS values (below 1.0) consistently occur when the weight on the Yield Strength goal is less than 0.20, regardless of other preferences.

11. A balanced weighting scheme at Level 1 does not produce a balanced outcome; it strongly favors DCI TS and DCI HV (values > 6.5) over EMI YS (value ~1.05).

12. The most extreme trade-offs occur when prioritizing Ferrite Fraction (Xf); for example, one scenario shows DCI Xf soaring to 13.0 while DCI Dα plummets to 0.79.

13. Shifting weight from Ferrite Fraction (Xf) to Pearlite Interlamellar Spacing (S0) at Level 2 can increase DCI S0 by over 50% while only causing a minor reduction in DCI Xf.

14. A high robustness for Ferrite Grain Size (DCI Dα > 3.0) is only achieved when the weight on the Dα goal is greater than 0.4.

15. Even a minimal weight (e.g., 0.06) on the Ferrite Fraction (Xf) goal is enough to push the DCI Xf value to its maximum if other Level 2 weights are also low.

16. The most robust solutions for DCI TS and DCI HV (values > 8.0) are found when Level 1 weights for both TS and HV are above 0.4.

17. It is impossible to achieve high robustness across all six goals simultaneously; at least one goal consistently falls to a lower value in every scenario.
18. Increasing the weight on the Yield Strength goal from ~0.33 to ~0.52 improves EMI YS by approximately 37% but reduces DCI TS by over 50%.
19. The DCI HV metric is generally less sensitive to trade-offs than DCI TS, often maintaining a higher value even when weights are shifted away from the HV goal.

These extracted statements can be treated as a knowledge layer in two complementary ways. First, they can remain as structured text linked to scenarios, forming a direct repository of insights. Second, they can be transformed into an LLM-based knowledge graph, as illustrated in Figure 6.2, where goals, decision variables, weights, and their connections are represented as interconnected nodes and relationships.

In the HRR test problem, we implement the schema-constrained knowledge-graph construction step using a fine-tuned open-weight language model. Specifically, we fine-tune a LLaMA-3.1-8B instruction-tuned model to convert validated natural-language knowledge statements into structured subject–relation–object triples. The fine-tuned model does not infer new relationships or learn engineering behavior; its role is limited to deterministic structural extraction using a predefined set of node types and relationship types.

The fine-tuning dataset consists of 100 input-output pairs, where each input is a synthetic but domain-consistent knowledge statement constructed using the controlled vocabulary and relationship types of the HRR test problem, and each output is one or more schema-compliant subject-relation-object triples. These statements were generated solely to support structural learning and were not derived from specific rc-cDSP scenarios. All pairs were manually checked to ensure relational correctness, numerical plausibility, and adherence to the predefined schema prior to fine-tuning. An illustrative example of the fine-tuning input-output format is shown in Table 6.8.

Table 6.8: Example Fine-Tuning Data for Schema-Constrained Knowledge-Graph Construction

```
{
  "input": "Convert the following engineering knowledge statement into knowledge-graph
triples using the predefined schema.
Statement: EMI_YS and DCI_TS exhibit a strong inverse relationship with a correlation of -
0.62.",
"output": {
   "triples": [
     {
       "subject": {"id": "EMI_YS", "type": "Goal"},
       "predicate": "negatively_correlates_with",
       "object": {"id": "DCI_TS", "type": "Goal"},
       "attributes": {"correlation": -0.62}
     }
   ]
  }
}
```

To clarify the structure of the LLM-based knowledge graph, we explicitly define its schema in terms of node types, relationship types, and associated attributes. The graph is composed of nodes representing goals across hierarchical levels, preference weights, and decision variables, which are connected through a fixed, predefined vocabulary of semantic relationships. Representative examples include correlation-based relations (e.g., *positively_correlates_with*, *negatively_correlates_with*), trade-off relations (e.g., *conflicts_with*, *synergizes_with*), and preference outcome relations (e.g., *weight_increases_goal*, *goal_is_sensitive_to_weight*). Selected relationships carry quantitative attributes, such as correlation magnitude, threshold values, percentage changes, or performance ranges, which numerically ground the extracted knowledge and support interpretable semantic querying. The complete set of node types, relationship types, and their associated attributes is summarized in Table 6.9. The knowledge graph

is constructed by transforming validated knowledge statements into structured triples using a fine-tuned large language model (LLaMA-3.1-8B-Instruct). The model is adapted using parameter-efficient LoRA fine-tuning (rank $r = 32$) and trained for three epochs using the AdamW optimizer with learning rate 2×10^{-4}. Inference is executed deterministically (temperature $= 0$) to ensure consistent schema-compliant extraction, building on an established LLM-based graph extraction approach [24]. The KG builder and the fine-tuned LLM are both available through the following GitHub repository[2].

Table 6.9: Knowledge Graph Relationship Types and Associated Attributes

Relationship type	**Attribute(s)**
positively_correlates_with	correlation_coefficient
negatively_correlates_with	correlation_coefficient
synergizes_with	correlation_coefficient or tradeoff_magnitude
conflicts_with	correlation_coefficient or tradeoff_magnitude
goal_is_sensitive_to_weight	value_range
goal_is_insensitive_to_weight	value_range
weight_increases_goal	threshold_value or percentage_change
weight_decreases_goal	threshold_value or percentage_change
belongs_to_level	no attribute (structural relation)

Although we illustrate the framework using the Hot Rod Rolling test problem, the proposed schema is domain-agnostic. The ontology defines general node categories (e.g., goals, decision variables, and preference weights) and relationship types (e.g., correlation, conflict/synergy, and weight-goal sensitivity) that are common across many multi-objective design problems. For other

[2] https://github.com/siavashmy/Knowledege_Graph.git

applications, the same relational structure can be retained while replacing the domain-specific variables and goals with those relevant to the new design context.

Using the created and quantitatively validated subject–relation–object triples, together with the schema-constrained knowledge-graph builder, we construct a knowledge graph for the HRR test problem. In Figure 6.2, node colors encode semantic roles within the decision framework: orange nodes represent preference weights, blue nodes represent system goals and decision variables, and green nodes denote system levels. The highlighted relationships illustrate how changes in preference weights propagate through the system. For example, increasing the weight on the Ferrite Fraction (Xf) objective is associated with a strong negative impact on the robustness of the DCI S_0 and DCI $D\alpha$ goals, with observed reductions on the order of 30–50% across the scenario set. By explicitly structuring such validated trade-offs as typed relationships with quantitative attributes, the knowledge graph supports semantic querying, pattern discovery, and interpretable reasoning over multi-level interactions, complementing and extending access beyond unstructured textual summaries, as detailed in Section 6.4.2.

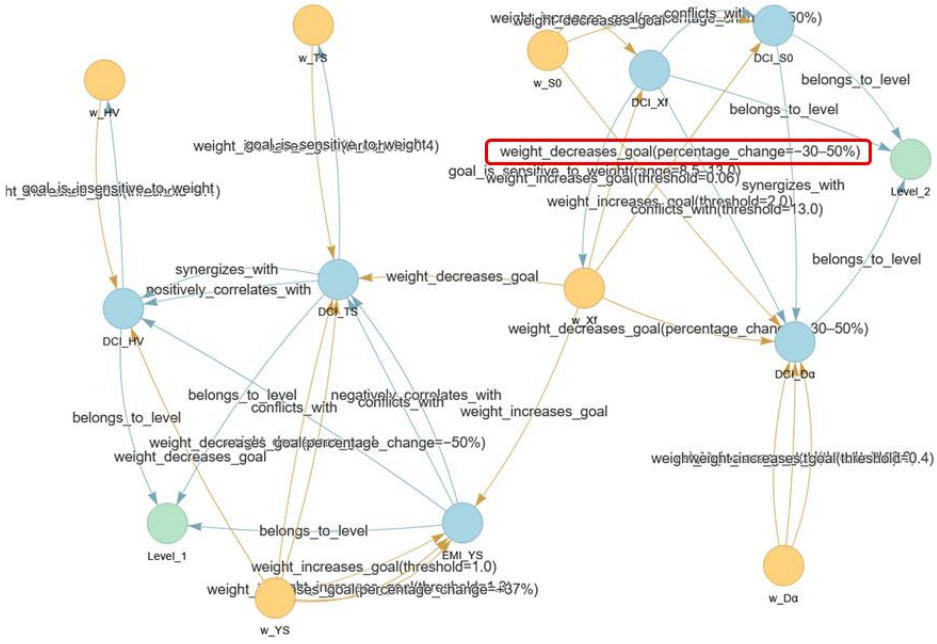


Figure 6.2: Knowledge Graph Representation of the Design Goals, Preference Weights, and Hierarchical Levels. Nodes represent goals, weights, and decision variables, while edges correspond to predefined relationship types, as described in Table 6.9.

6.4.2 Knowledge Exploration (Step 2)

Knowledge exploration through querying and clustering is essential in the HRR test problem because it converts the high-dimensional scenario data into interpretable patterns. By highlighting targeted trade-offs and grouping scenarios into strategic archetypes, these methods reveal non-intuitive relationships and provide designers with actionable insights that would be difficult to identify through manual inspection.

6.4.2.1 Interactive Querying (Step 2.1)

Through the proposed framework, designers are given the opportunity to interactively query the model, enabling them to explore scenarios and trade-offs more effectively. This interactive capability supports deeper understanding and aids in guiding design decisions. In this section, we discuss the results obtained through such querying. In these queries, the knowledge graph is first used to interpret user intent by identifying relevant entities, relationships, and associated quantitative attributes, which then guide numerical retrieval over the structured scenario dataset.

Query 1: *"Which scenarios provide the highest DCI TS (Tensile Strength) while keeping EMI YS (Yield Strength) above the minimum robustness threshold of 1.0?"*

AI Response: The best candidates that balance high DCI TS with an acceptable EMI YS are:

- Scenario #55: Provides a DCI TS of 6.82 while maintaining an EMI YS of 1.078.
- Scenario #29: Offers a DCI TS of 6.65 with an EMI YS of 1.070.

These scenarios represent a strong compromise, successfully avoiding the extremely low DCI TS values typically seen in solutions with high EMI YS.

Query 2: *"What is the best way to improve both DCI Dα (Ferrite Grain Size) and DCI S0 (Pearlite Interlamellar Spacing) at the same time?"*

AI Response: These two goals are highly synergistic, with a strong positive correlation of +0.750. To improve both, a designer should increase the weights on the S0 (Interlamellar Spacing) and Dα (Ferrite Grain Size) goals at Level 2, while reducing the weight on the Xf (Ferrite Fraction) goal. Scenario #93 is a prime example of this strategy. With high weights on S0

= 0.42 and Dα = 0.35, it achieves one of the highest DCI Dα = 3.17 and the highest DCI S0 = 11.54 in the dataset.

The following query further illustrates how the knowledge graph supports intent-aware reasoning by explicitly accounting for known trade-offs and sensitivity relationships before retrieving numerical scenarios. The knowledge graph is first used to interpret the designer's intent by identifying the relevant goals (EMI_YS and DCI_TS), their hierarchical levels, and the known trade-off and sensitivity relationships between them. This semantic interpretation defines a retrieval plan with a minimum EMI_YS threshold and limited degradation in DCI_TS, which is then applied to the structured scenario dataset to filter and rank candidate solutions.

Query 3: *"If a designer wants to achieve a high EMI YS but avoid severe degradation in DCI TS, which preference adjustments are most appropriate, and what trade-offs should be expected?"*

AI Response: The results show that excessive prioritization of Yield Strength significantly degrades DCI_TS due to the documented trade-off. A moderate increase in the Yield Strength weight achieves EMI_YS above the robustness threshold while limiting tensile strength loss. Scenario #55 represents this balance, demonstrating how the knowledge graph enables intent-aware and interpretable decision support prior to numerical scenario selection.

6.4.2.2 Scenario Clustering and Narratives (Step 2.2)

The 169 scenarios can be grouped into two distinct design archetypes. Each cluster represents a different strategic approach to solving the design problem.

Cluster A: The "High Performance" Archetype

- ***Characteristics:*** This group features the highest DCI TS and DCI HV values but consistently shows the lowest EMI YS.
- ***Narrative:*** "These scenarios represent solutions that prioritize maximum mechanical performance. They produce the strongest and hardest steel rods but are less robust to potential inaccuracies in the Yield Strength model. This is the ideal cluster if achieving top-tier performance metrics is the primary concern and model uncertainty is a secondary issue."

Cluster B: The "Microstructure-Focused" Archetype

- ***Characteristics:*** This cluster contains the most extreme values for the Level 2 goals, featuring both the highest and lowest results for DCI Dα (Ferrite Grain Size), DCI S0 (Pearlite Interlamellar Spacing), and DCI Xf (Ferrite Fraction).
- ***Narrative:*** This group represents design strategies driven primarily by microstructure targets rather than final product performance. It includes highly specialized solutions, such as those that maximize Ferrite Fraction (Xf) at all costs. While these scenarios are not balanced, they are critical for understanding how Level 2 process decisions on microstructure directly propagate to system-wide outcomes, often creating strong conflicts with product-level properties.

6.4.3 Adaptable Solution-Space Generation (Step 3)

The focus in the final component of the proposed framework is on adaptability for solution generation, ensuring that the decision-making process remains flexible and responsive to evolving designer preferences. When thresholds are applied, the result may be an empty feasible set, leaving designers uncertain about how to proceed. The adaptable solution-space generation step addresses this challenge by using the generative AI to filter scenarios against the thresholds and, crucially, to explain the outcome. If no solutions exist, the system provides targeted recommendations on which thresholds could be relaxed, and by how much, to restore feasibility. Here, the LLM acts as an adaptive guide, iteratively updating the feasible solution set as thresholds are relaxed or tightened. Designers can thus converge toward satisficing solutions through a dialogue with the AI system. We now present three examples to further clarify this process.

Example 1: No Feasible Solution under Initial Thresholds

In the first configuration (Figure 6.3), the designer sets thresholds across all six goals: EMI YS ($G_1 \geq 1.3$), DCI TS ($G_2 \geq 6$), DCI HV ($G_3 \geq 6$), Dα ($G_4 \geq 2$), So ($G_5 \geq 9$), and Xf ($G_6 \geq 10$). When these requirements are applied simultaneously, the result is an empty feasible set, no scenario satisfies all thresholds. This outcome highlights the fundamental trade-offs among the goals, especially the conflict between EMI YS and DCI TS/HV. In traditional

methods, the process would typically stop at this point, leaving designers to manually adjust their preferences. In contrast, the proposed framework allows the generative-AI engine to interpret the infeasibility and provide guidance.

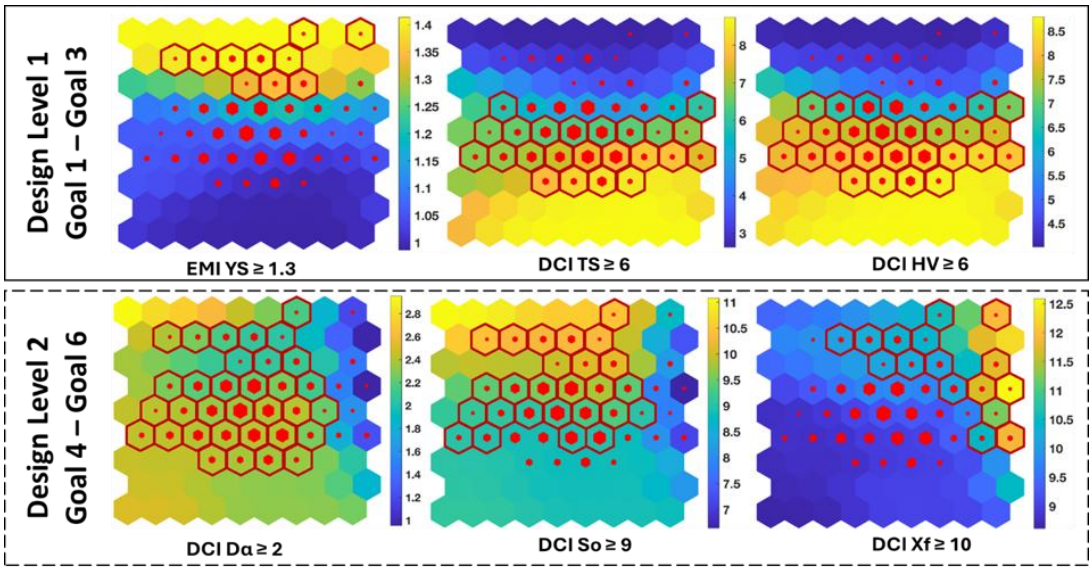


Figure 6.3: Preliminary iSOM plots for goals at design levels 1 and 2, where yellow and dark blue areas correspond to regions of relatively high and low robustness, respectively. Hexagonal iSOM grid points marked in red denote regions of satisficing solutions for the goals. The red dots represent the design scenarios mapped onto the iSOM grid points.

Specifically, it identifies the primary conflicts and suggests candidate adjustments to restore feasibility, pointing to the need for relaxing selected thresholds. These insights are generated by analyzing trade-off patterns across the scenario repository together with relationships captured in the knowledge graph, allowing the LLM to identify conflicting goals and propose candidate threshold relaxations. The conversations are shown as "Model Input/Output" in Table 6.10.

Table 6.10: LLM Input-Output Showing No Feasible Solution

Model Input
"Filter the 169 design scenarios and return only those that satisfy the following thresholds: EMI YS ≥ 1.3, DCI TS ≥ 6, DCI HV ≥ 6, DCI Dα ≥ 2, DCI So ≥ 9, and DCI Xf ≥ 10. If no scenarios satisfy all these conditions, explain why and suggest which thresholds could be relaxed to restore feasibility."
Model Output
"I checked all 169 scenarios against your thresholds. ***No scenario satisfies*** *all six requirements simultaneously.* *The main conflict lies between EMI YS and the mechanical performance goals DCI TS and DCI HV. Scenarios that achieve high EMI YS (≥ 1.3) consistently fall below the required thresholds for TS and HV (typically around 3-4, not ≥ 6).* *To restore feasibility, you have two options:* *1- Relax EMI YS (e.g., remove it as a strict constraint, or lower to ~1.0–1.1).* ***2- Reduce the requirement on DCI TS from 6.0 to ~3.5, and DCI HV from 6.0 to ~5.0****.* *With these adjustments, feasible solutions appear, particularly in scenarios like #40-#44 and #92-#95, which maintain robust values for DCI* So *(≥ 10) and DCI Dα (≥ 10) while offering moderate performance trade-offs.*

Example 2: Restoring Feasibility through Relaxed Thresholds

In the second configuration (Figure 6.4), thresholds are strategically relaxed. EMI YS (G_1) is excluded as a strict constraint, while DCI TS (G_2) is reduced from 6.0 to 3.5, and DCI HV (G_3) is reduced from 6.0 to 5.0. The remaining thresholds - Dα ($G_4 \geq 2$), So ($G_5 \geq 9$), and Xf ($G_6 \geq 10$) - are retained. Under these adjusted satisficing limits, feasible solutions emerge, demonstrating that the system can recover a valid solution space once conflicting requirements are moderated. Importantly, LLM does not simply return the feasible set but also explains the underlying trade-offs. This guidance helps designers understand not just where feasible solutions exist, but why relaxing specific thresholds makes them attainable. In Table 6.11 and 6.12, we present the common satisficing solutions identified across Design Levels 1 and 2 through AI-based exploration after relaxing the satisficing limits.

Table 6.11: LLM Input-Output for Identifying Satisficing Solutions (Relaxed Thresholds)

Model Input
"Update the thresholds as follows: EMI YS ≥ 1.3, DCI TS ≥ 3.5, DCI HV ≥ 5.0, DCI Dα ≥ 2, DCI So *≥ 9, and DCI Xf ≥ 10. Which scenarios are now feasible, and what trade-offs do they represent?"*
Model Output
"With the updated thresholds, several scenarios now satisfy the requirements. Representative feasible candidates include: *Scenario 40: EMI YS = 1.45, DCI TS = 3.25, DCI HV = 4.62, DCI Dα = 2.00, DCI* So *= 10.40, DCI Xf = 10.21.* *Scenario 42: EMI YS = 1.45, DCI TS = 3.25, DCI HV = 4.62, DCI Dα = 1.94, DCI* So *= 7.96, DCI Xf = 11.95.* *Scenario 93: EMI YS = 1.38, DCI TS = 2.37, DCI HV = 3.79, DCI Dα = 3.17, DCI* So *= 11.54, DCI Xf = 8.91.* *These scenarios demonstrate that relaxing the TS/HV thresholds restores feasibility. The trade-off is clear:* *2- You gain higher robustness in EMI YS and microstructure goals (*So*, Xf, Dα).* *1- You must accept lower tensile strength and hardness (TS, HV values ~3–5).* ***Interpretation**: Relaxing TS and HV requirements allows designs that emphasize reliability (high EMI YS) and strong microstructural robustness (DCI* So *and Xf). This trade-off is valuable if stability and material consistency are prioritized over maximum mechanical performance."*

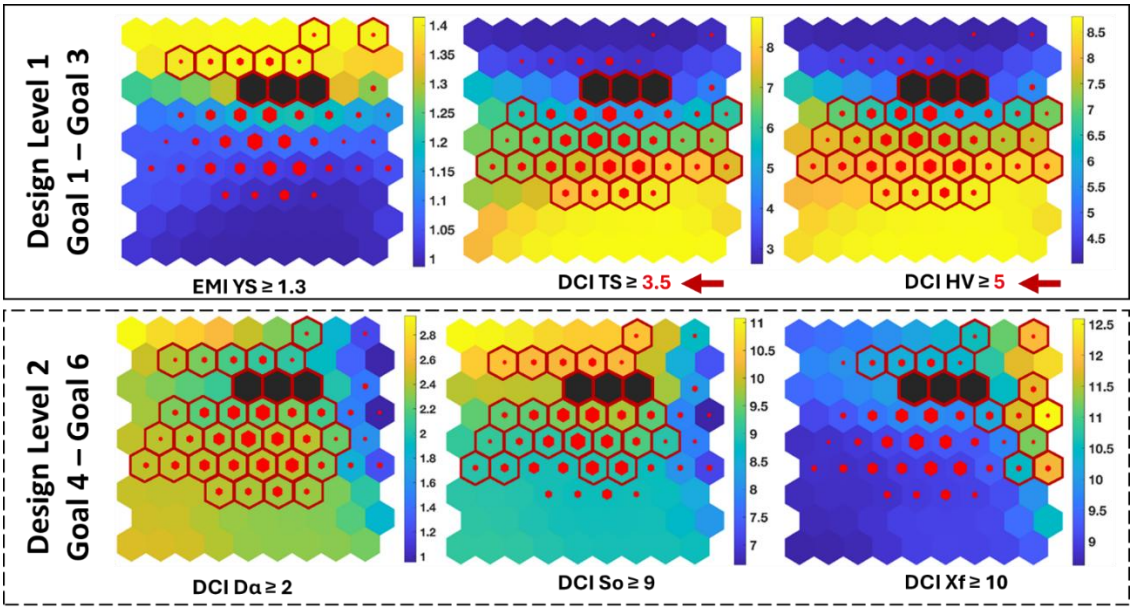


Figure 6.4: iSOM plots for all goals across design levels 1 and 2 after updating the satisficing goal thresholds. Red-highlighted iSOM grid points indicate the satisficing solution regions for individual goals, while black grids represent the common satisficing solution across all goals and design levels.

Table 6.12: Common Satisficing Solutions Identified Across Design Levels 1 and 2 Using AI Exploration After Relaxing the Satisficing Limits

Design	**Design Level 1 Goals**			**Design Level 2 Goals**		
Scenario #	EMI YS	DCI TS	DCI HV	DCI Dα	DCI So	DCI Xf
40	1.45	3.25	4.62	2.00	10.40	10.21
42	1.45	3.25	4.62	1.94	7.96	11.95
93	1.38	3.37	3.79	3.17	11.54	8.91

Example 3: Iterative Adaptation and Designer Control

In this example, we demonstrate how iterative adaptation enables designers to shift priorities without losing oversight of the decision process. Initially, strict Level-1 thresholds (e.g., DCI TS ≥ 6.0, DCI HV ≥ 6.0) excluded all solutions because we observe a strong conflict between mechanical performance and microstructure robustness in the dataset.

By interactively relaxing the TS and HV requirements and rebalancing Level-1 weights toward YS, the feasible set reappeared, providing new satisficing options that meet Level-2 robustness goals. Importantly, the LLM does not simply list feasible scenarios; it also explains why the feasible set changes (e.g., lower mechanical strength enables more robust microstructure) and what trade-offs must be accepted. If priorities shift again, for example, if the next product release demands higher mechanical performance, the same process allows the designer to tighten thresholds, immediately observing which scenarios are excluded and how the solution landscape narrows. This back-and-forth process illustrates our notion of iterative adaptation: designers can explore 'what-if' scenarios, progressively refine requirements, and remain in control of the decision-making process. The role of the AI is to generate transparent, narrative explanations of evolving trade-offs, ensuring that the search for satisficing solutions remains interpretable and aligned with human judgment. We present the model inputs and outputs for this case below. In Table 6.13 and 14, we demonstrate the updated satisficing solutions identified across Design Levels 1 and 2.

Table 6.13: LLM Input-Output for Identifying Updated Satisficing Solutions

Model Input
" For the upcoming product release, our priority has shifted toward robustness at Design Level 2 (microstructure). We need solutions where DCI Dα ≥ 2.0, DCI So ≥ 9.0, and DCI Xf ≥ 10.0. Given this new focus, how should we adjust our Level-1 targets (EMI YS, DCI TS, DCI HV) and weight settings for YS/TS/HV so that these robust Level-2 solutions become feasible? Please also point out which scenarios from the dataset meet these updated requirements and explain the trade-offs we would face compared to our previous, more performance-oriented goals?
Model Output
"To prioritize Level-2 robustness, I recommend relaxing Level-1 thresholds and shifting weight allocations:
Level-1 thresholds:
• EMI YS ≥ 1.30
• DCI TS ≥ 3.25 (reduced from 6.0)

• DCI HV ≥ 4.62 (reduced from 6.0)

Weight allocations:

• YS ≈ 0.45–0.55 (increase emphasis)

• TS ≈ 0.20–0.30 (decrease)

• HV ≈ 0.15–0.30 (moderate)

Feasible scenarios under these settings include:

Scenario 40 — EMI YS = 1.45, TS = 3.25, HV = 4.62, Dα = 2.00, So = 10.40, Xf = 10.21.

Scenario 49 — EMI YS = 1.45, TS = 3.25, HV = 4.62, Dα = 2.00, So = 10.40, Xf = 10.21.

Scenario 52 — EMI YS = 1.45, TS = 3.25, HV = 4.62, Dα = 2.00, So = 10.40, Xf = 10.21.

Scenario 144 — EMI YS = 1.45, TS = 3.25, HV = 4.62, Dα = 2.00, So = 10.40, Xf = 10.21.

Trade-off explanation: *These solutions achieve robust microstructure metrics while maintaining acceptable EMI YS. The trade-off is a moderate drop in mechanical performance: TS and HV remain around 3.25 and 4.62 instead of the higher values in other designs."*

Table 6.14: Updated Satisficing Solutions Identified Across Design Levels 1 and 2

Design	**Design Level 1 Goals**			**Design Level 2 Goals**		
Scenario #	EMI YS	DCI TS	DCI HV	DCI Dα	DCI So	DCI Xf
40	1.45	3.25	4.62	2.00	10.40	10.21
49	1.45	3.25	4.62	2.00	10.40	10.21
52	1.45	3.25	4.62	2.00	10.40	10.21
144	1.45	3.25	4.62	2.00	10.40	10.21

6.4.4 Results and Discussion

Through the test problem in Section 6.3, we demonstrate how the framework uncovers non-intuitive trade-offs and identify robust satisficing solutions that balance conflicting objectives. For example, strict targets for tensile strength and hardness (Level-1) initially excluded all feasible scenarios; as we relaxed the thresholds iteratively, we found that scenarios weighting yield strength more heavily reach higher Level-2 robustness. We used the framework to highlight scenarios such

as 40, 49, 52, and 144 and we provided narrative explanations of why these tradeoffs emerged. Importantly, when comparing the outcomes generated using the proposed generative-AI framework with those obtained from the iSOM results in the source study [7], we observe strong consistency in the identification of feasible solutions, trade-offs, and scenario interpretations. In other words, the generative-AI framework successfully reproduces the insights provided by iSOM in this specific test problem, including:

i. The identification of infeasible regions when strict thresholds are applied.

ii. The process of threshold relaxation to restore feasibility.

iii. The connection and interactions between Design Level 1 (materials) and Design Level 2 (cooling/process) objectives.

This observed consistency demonstrates that the generative-AI approach not only reproduces the interpretability of iSOM but also extends it by offering dynamic, interactive, and narrative explanations of the evolving solution space. As a result, designers can explore "what-if" scenarios, adapt priorities in real time, and remain in control of decision-making, while benefiting from insights that are both interpretable and aligned with established visualization methods.

LLM-based outputs can be sensitive to prompt phrasing, may hallucinate unsupported statements, and may present overconfident or internally inconsistent explanations, risks that are especially important in high-stakes engineering decision support. In this study, we mitigate these risks by running the model deterministically (temperature = 0) and constraining the agent to operate over the fixed rc-cDSP scenario repository and its derived knowledge graph, so responses are grounded in the available data rather than unconstrained generation. We emphasize that LLM narratives are decision aids and should be cross-checked against the underlying scenario records before final design actions.

6.5 Closing Remarks for Chapter 6

In this chapter, we introduce a Generative AI framework, augmented by knowledge graphs, to support interpretable satisficing solution design in complex multi-objective manufacturing contexts. Using the Hot Rod Rolling (HRR) PMMP system as a testbed, we demonstrate the framework's capability to go beyond static visualization by offering dynamic, human-centered

decision support. Although demonstrated on the HRR-PMMP system, the framework is generalizable to other complex multi-objective design problems across diverse domains.

Through the results, we underscore three key outcomes:

i. ***Interpretability:*** Large Language Models (LLMs) effectively summarize complex trade-offs, generate narrative explanations of scenario clusters, and guide designers through infeasible regions with transparent reasoning.

ii. ***Adaptivity:*** Designers can use the framework to dynamically update feasible solution spaces as designer preferences evolve, enabling iterative exploration of "what-if" scenarios while maintaining decision-maker control. The adaptivity demonstrated here is adaptive human-in-the-loop exploration of an existing rc-cDSP solution space, enabling iterative preference-driven feasibility restoration and interpretable trade-off navigation; extending this to adaptive solution generation through new rc-cDSP evaluations is a clear and intended direction for future work.

iii. ***Reliability:*** Insights derived through GenAI are consistent with established visualization tools like iSOM, while also enhancing computational efficiency, transparency, and usability in high-dimensional environments.

In the proposed framework we operate in two computational phases. In the offline phase, knowledge statements are extracted from the scenario repository and used to construct the knowledge graph. This step is performed once for the dataset and its computational cost scales with the number of scenarios processed. In the online phase, designers interactively query the structured knowledge graph to retrieve relevant relationships and supporting scenario evidence for decision-making. This shifts the scalability bottleneck away from human visual bandwidth which can degrade under clutter and many-objective complexity in traditional visualization workflows; toward retrieval and summarization pipelines that can be engineered to handle orders-of-magnitude larger datasets.

Open Research Issues: We demonstrate the promise of GenAI in multi-objective designing, but several open challenges remain that are critical for its evolution. We need to embed domain-specific engineering knowledge into LLMs to move beyond generic reasoning toward technically rigorous guidance. We also need to develop physics-consistent and multimodal generative models to ensure validity across numerical, graphical, and simulation-driven data. Scaling the framework to industrial-size problems will require advances in computational efficiency and methodical

innovation. Another promising direction is the development of quantitative metrics to evaluate how faithfully LLM-generated narratives represent the statistical properties of clustered design scenarios, ensuring that archetypal descriptions remain consistent with the underlying data distributions. Finally, we must strengthen transparency, explainability, and trust so that GenAI can be adopted as a dependable partner in high-stakes engineering decision-making. By addressing these issues, the research community can advance GenAI from a promising prototype into a transformative force in human-centered multi-objective designing. Building on the work presented in this chapter, we invite researchers to further develop these directions and explore following open research questions.

- *How can we design a conversational, designer-in-the-loop workflow that closes the adaptivity gap by updating satisficing guidance on-the-fly as thresholds and preference priorities change, while logging provenance (data subset used, query/prompt, and supporting scenario evidence) for each narrative explanation in the knowledge graph?*
- *How can we integrate a lightweight generative sampler with rc-cDSP to proactively expand the design space, that is, , propose new preference-weight vectors and/or decision-variable seeds in unexplored or infeasible regions, evaluate them via rc-cDSP under uncertainty across levels, and then use LLM narratives to critique results and steer the next sampling step in a closed loop?*

We opened this chapter by setting out to address RQ3a and to examine Hypothesis 3a. Within the scope of the Hot Rod Rolling manufacturing problem we study, the evidence we present is consistent with that hypothesis: a generative AI framework in which a large language model extracts knowledge statements from structured scenario data, each statement is validated quantitatively before it is admitted to the Knowledge Graph, and the validated graph supports interactive querying and infeasibility recovery, allows us to navigate a high-dimensional satisficing space with interpretable guidance whose conclusions are consistent with established visualization tools such as iSOM. On that basis we offer the new knowledge we set out to produce: a GenAI-anchored Knowledge Graph framework for augmenting E-CPSS design decision-making, with LLM-based knowledge extraction, quantitative validation protocols, and interactive trade-off summarization. We take this to contribute the Augmented Intelligence layer of the DSIM, and the Knowledge Graph it establishes is the structure on which the autonomous agents of the next

chapter draw. We note the scope of the present demonstration plainly: the adaptivity we show is interactive, human-in-the-loop exploration of an existing solution space, and extending it to adaptive solution generation through new evaluations remains future work. With respect to verification and validation, we read the Frame of Reference as offering theoretical structural validity, the Hot Rod Rolling problem as a reasonable vehicle for empirical structural validity, and the demonstrated interpretability, adaptivity, and consistency with established tools as empirical performance validity within the bounds of this single problem; theoretical performance validity we defer to Chapter 9.

The methods to this point keep the designer in the loop, working at human speed: we query, we interpret, we act. Where the system affords time to think before it reaches its next state, that pace carries no penalty. What we do not yet address is the setting in which the system does not wait, in which the rules themselves shift mid-operation and a response is required before the next decision arrives. That is the work of Chapter 7, where we build the capability to act autonomously at the timescale of the system, drawing on the Knowledge Graph established here.

References

[1] Lotov, A. V., and Miettinen, K., 2008, "Visualizing the Pareto Frontier," *Multiobjective Optimization*, J. Branke, K. Deb, K. Miettinen, and R. Słowiński, eds., Lecture Notes in Computer Science, Vol. 5252, Springer, Berlin, Heidelberg.

[2] Simon, H. A., 1996, *The Sciences of the Artificial*, MIT Press.

[3] Mistree, F., Smith, W., and Bras, B., 1993, "A Decision-Based Approach to Concurrent Design," *Concurrent Engineering*, H. R. Parsaei and W. G. Sullivan, eds., Springer, Boston, MA.

[4] Bras, B., and Mistree, F., 1993, "Robust Design Using Compromise Decision Support Problems," *Engineering Optimization*, 21(3), pp. 213–239.

[5] Chen, W., Allen, J. K., Mistree, F., and Tsui, K. L., 1996, "A Procedure for Robust Design: Integrating Taguchi's Parameter Design Approach With Optimization Under Uncertainty," *Journal of Mechanical Design*, 118(4), pp. 478–485.

[6] Filipič, B., and Tušar, T., 2018, "A Taxonomy of Methods for Visualizing Pareto Front Approximations," *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 649–656.

[7] Nagar, D., Ramu, P., and Deb, K., 2021, "Interpretable Self-Organizing Maps (iSOM) for Visualization of Pareto Front in Multiple Objective Optimization," *International Conference on Evolutionary Multi-Criterion Optimization*, Springer International Publishing, Cham, Switzerland, pp. 645–655.

[8] Baby, M., Rama Sushil, R., Ramu, P., Allen, J. K., Mistree, F., and Nellippallil, A. B., 2024, "Robust, Co-Design Exploration of Multilevel Product, Material, and Manufacturing Process Systems," *Integrating Materials and Manufacturing Innovation*, 13(1), pp. 14–35.

[9] Balaji, N., Baby, M., Sharm, and Ramu, P., 2026, "Framework for Systematic and Efficient High-Dimensional Robust Design Exploration – A Composite Materials Example," *International Journal on Interactive Design and Manufacturing*, 20, pp. 61–76.

[10] Shneiderman, B., 2020, "Human-Centered Artificial Intelligence: Reliable, Safe and Trustworthy," *International Journal of Human–Computer Interaction*, 36(6), pp. 495–504.

[11] Regenwetter, L., Srivastava, A., Gutfreund, D., and Ahmed, F., 2023, "Beyond Statistical Similarity: Rethinking Metrics for Deep Generative Models in Engineering Design," *Computer-Aided Design*, 165, p. 103609.

[12] Yu, C., Zheng, P., Peng, T., Xu, X., Vos, S., and Ren, X., 2025, "Design Meets AI: Challenges and Opportunities," *Journal of Engineering Design*, 36(5–6), pp. 637–641.

[13] Tušar, T., and Filipič, B., 2014, "Visualization of Pareto Front Approximations in Evolutionary Multiobjective Optimization: A Critical Review and the Prosection Method," *IEEE Transactions on Evolutionary Computation*, 19(2), pp. 225–245.

[14] Chakraborty, T., Wirth, C., and Seifert, C., 2025, "Comparative Explanations: Explanation Guided Decision Making for Human-in-the-Loop Preference Selection," *World Conference on Explainable Artificial Intelligence*, Springer Nature Switzerland, Cham, pp. 139–161.

[15] Duan, R., Karthik, N., Chen, Y., Shi, J., Jain, R., Yang, M., and Ramani, K., 2026, "ConceptVIS: Exploring and Visualizing Design Concepts With Large Language Models

Using Interactive Knowledge Graph," *Journal of Computing and Information Science in Engineering*, 26(1), p. 011002.

[16] Gupta, H., Bhalerao, M. J., Honeycutt, W. T., Allen, J. K., and Mistree, F., 2025, "On Designing Evolving Cyber–Physical–Social Systems: A Decision-Based Design Perspective," *Journal of Computing and Information Science in Engineering*, 25(12), p. 120803.

[17] Deb, K., Sinha, A., Korhonen, P. J., and Wallenius, J., 2010, "An Interactive Evolutionary Multiobjective Optimization Method Based on Progressively Approximated Value Functions," *IEEE Transactions on Evolutionary Computation*, 14(5), pp. 723–739.

[18] Afsar, B., Ruiz, A. B., and Miettinen, K., 2023, "Comparing Interactive Evolutionary Multiobjective Optimization Methods With an Artificial Decision Maker," *Complex and Intelligent Systems*, 9(2), pp. 1165–1181.

[19] Miettinen, K., and Hakanen, J., 2017, "Why Use Interactive Multi-Objective Optimization in Chemical Process Design?" *Multi-Objective Optimization*, pp. 157–197.

[20] Bhalerao, M. J., Mandegari, S., Allen, J. K., and Mistree, F., 2025, "Designing Robust Policies for Evolving Cyber-Physical-Social Systems in Uncertain Environments," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, Vol. 89213, p. V02BT02A059.

[21] Peckham, O., Raines, J., Bulsink, E., Goudswaard, M., Gopsill, J., Barton, D., Nassehi, A., and Hicks, B., 2025, "Artificial Intelligence in Generative Design: A Structured Review of Trends and Opportunities in Techniques and Applications," *Designs*, 9(4), p. 79.

[22] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A. M. T., 2002, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, 6(2), pp. 182–197.

[23] Kretzschmar, M., Dammann, M. P., Schwoch, S., Braun, F., Saske, B., and Paetzold-Byhain, K., 2024, "Evaluating the Role of Generative AI in Product Development and Design-A Systematic Review," *DS 130: Proceedings of NordDesign 2024*, Reykjavik, Iceland, pp. 21–30.

[24] Zhang, B., and Soh, H., 2024, "Extract, Define, Canonicalize: An LLM-Based Framework for Knowledge Graph Construction," *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9820–9836.

CHAPTER 7

AUTONOMOUS STRATEGY EXECUTION: KNOWLEDGE-GOVERNED AGENTS FOR ADAPTIVE DECISION-MAKING IN E-CPSS

Background and Context

By the close of Chapter 6, we can read the state of an E-CPSS, structure robust decisions within it, design adaptive strategies across its transitions, and augment our own judgment when the design space grows past what we can navigate unaided. In each of these, we keep the designer in the loop and working at human speed: we query, we interpret, we act. For agricultural policy cycles and multi-decade sustainability planning, we can work at that pace without penalty, because we have time to think before the system reaches its next state. In this chapter, we build the capability to act when the system does not wait.

The condition we take up is one in which the rules themselves shift mid-operation. A regulator raises a penalty and the cost of inaction triples. A supplier disrupts the cost of delay quintuples. The goals that were appropriate at the start of an operating period are no longer appropriate by its end, and we must answer the change before the next decision arrives, without pausing to reconvene, reformulate, and rerun. Across supply chains, financial workflows, energy balancing, and regulated production schedules, we face the same property: we are exposed to ongoing change that is signal rather than noise. At this pace, we cannot treat autonomous agents as a convenience we add for efficiency. We need them as a structural requirement of the design problem itself.

We meet this requirement through RQ3b, and it is where we locate the open research gap. When we hand execution to a learning agent, we appear to risk the very properties we establish in the previous chapters, because an agent that learns its own policy can drift from robustness, adaptivity, and interpretability the moment the world it trained on shifts. We frame RQ3b around this gap and state our hypothesis directly: an agent that draws its standard for acceptable performance from a structured, dynamically updated description of the world maintains satisficing performance and

adapts more reliably under non-stationary E-CPSS conditions than an agent that learns from the reward signal alone, without surrendering the guarantees a high-stakes setting requires.

We shape our answer by carrying the satisficing philosophy of Chapters 4 and 5 into this regime. The agent does not pursue a global optimum that no longer exists once conditions move; it pursues a standard for what counts as good enough. Our contribution in this chapter is to make that standard responsive to the world rather than fixed at design time. We draw the agent's aspiration from the structured and evolving Knowledge Graph that we first established in Chapter 6, and we hold the channel through which the agent learns its values apart from the channel through which it forms its aspiration. By keeping the two separate, we let the agent follow a change in the world without corrupting what it has already learned, and we let it recover cleanly when that change is later withdrawn. We hold ourselves to four requirements a high-stakes setting cannot do without: we preserve the convergence that makes value learning trustworthy, guarantee the quality of decisions taken before convergence, stay robust when the structured information is wrong, and remain auditable down to the specific facts that drove each decision. We demonstrate the framework on supply chains, where we treat non-stationarity not as an occasional disturbance but as the defining condition.

Through this work, we contribute $M(s_i)$ and $K(s_i)$ to Layer 4 of the Decision-State Information Model, the Execution layer. We offer $M(s_i)$ as the autonomous execution method, the means by which we carry strategies out at the timescale of the system rather than the timescale of the designer, and $K(s_i)$ as the experiential knowledge the agent accumulates as it acts, which we hold in the Knowledge Graph as structured, queryable relationships and feed forward into the next state. In building this layer, we operationalize three E-CPSS design requirements from Chapter 2: Self-Adaptability, through aspirations we update as the world changes; Decentralized Decision-Making, through agents that decide without waiting on a central authority; and Self-Organizability, through behavior we hold within governance limits by the structure of the method rather than by supervision. Beneath the results lies the commitment we carry through the DSIM as a whole: we do not treat what counts as good enough as a constant of the problem but as a variable, recomputed from the changing structure of the world, so that the objective stays coupled to the system rather than frozen at design time.

One dimension stays untouched at the close of this chapter, and we recognize it as the one on which everything finally turns. Everything here, we evaluate against technical performance and governance rules. What we do not yet measure is how the people on the receiving end respond. The strategies we execute eventually reach farmers, communities, workers, and businesses, who answer not through reward functions but through cognition, emotion, memory, and social dynamics that rational-actor models cannot represent. Whether we succeed in the world depends on that human response, and we take up the work of assessing it before deployment in Chapter 8.

Preface

Many decision settings change while the system is operating. Conditions shift during an operating period, and the goals that were appropriate at its start no longer fit by its end. A decision-maker has to respond before the next decision arrives, without pausing the world to gather data and retrain. The methods we usually reach for are built for a fixed environment and a stationary objective, so when conditions change in structure rather than in noise, the agent keeps pursuing a goal that no longer fits the world it faces. It then either adapts slowly as evidence accumulates or waits for a human to intervene, and in the settings where we most need reliable automation we can least afford either. We take up a single question: once the world has changed, what should an agent treat as good enough, and how can it act on that judgment with guarantees we can trust in a high-stakes setting?

We give such an agent a principled way to revise what counts as good enough as the world changes, while preserving the behavior that makes a learned policy safe to field. We bring Herbert Simon's satisficing principle into reinforcement learning by drawing the aspiration, the agent's standard for what counts as good enough, from a structured and evolving description of the world rather than fixing it in advance by hand. We introduce the Compiled-Aspiration Markov Decision Process and an algorithm, CA-RS-Q, to realize this idea, and we keep the channel through which the agent learns its values separate from the channel through which it forms its aspiration. By holding the two channels apart we let the agent follow changes in the world without corrupting what it has learned, and we let it recover cleanly when we later correct or withdraw a change.

We set four requirements that we treat as non-negotiable for deployment in a high-stakes environment. Through the method we must preserve the convergence that makes value learning trustworthy, guarantee the decisions we take before convergence, stay robust to error in the

structured information we depend on, and remain auditable. We prove that the agent's value estimates still converge once the world settles; that the actions we select meet the aspiration on a meaningful fraction of decisions while learning is still under way; and that the value of the policy degrades only gracefully when the structured information is wrong. We are not aware of a directly comparable robustness guarantee, and we offer it as a contribution rather than a settled result. Auditability we build in as a structural property of the method rather than add as a feature after the fact.

We examine the framework on three inventory benchmarks that span a wide range of scale, run under identical settings, and we find greater effectiveness at larger scale rather than less. Under a regulator shock that abruptly raises the cost of failure, the agent compiles the change into its aspiration and produces far fewer catastrophic outcomes than a standard learner. We make no claim to an advantage on raw return; in stationary, well-specified regimes the agent is deliberately conservative, and we accept that cost, because the four properties we establish cannot be had from return maximization alone. We can trace every decision back to the specific facts we used to set the aspiration, and we demonstrate this on a single worked decision a reviewer can follow line by line.

We draw three conclusions. The aspiration in satisficing reinforcement learning need not be arbitrary; we can take it from the structured information through which we describe the operating environment. We need not trade convergence, a guarantee before convergence, robustness, and auditability against one another; they coexist when we keep the learning channel and the aspiration channel apart. And we can make auditability survive the algorithm rather than reconstruct it afterward, if we design the decision structure for it from the start.

Beneath these lies a more fundamental point. What an agent should regard as good enough is not a constant of the problem but a variable, and we recompute it from the changing structure of the world. In the environments where complex, high-stakes systems actually operate, settings that evolve in structure rather than merely in noise, there is no fixed optimum to settle into, because the conditions an agent acts under keep changing. By compiling the aspiration from a living description of those conditions, we let an agent's standard for a good decision co-evolve with the system it operates in, so that the objective stays coupled to the world rather than frozen at design time. We offer this coupling, of what counts as good enough to the present structure of a changing system, as the more general contribution that the specific results above make concrete.

Keywords: reinforcement learning, satisficing, knowledge graph, bounded rationality, interpretability, contextual Markov decision process, inventory management

Research Question	RQ3b. What are the principles and method for enabling an autonomous agent to execute adaptive strategies in an E-CPSS under non-stationary conditions, while preserving the robustness, adaptivity, and interpretability established in the preceding chapters?
Hypothesis	Hypothesis 3b. An agent that draws its standard for acceptable performance from a structured, dynamically updated description of the world maintains satisficing performance and adapts more reliably under non-stationary E-CPSS conditions than an agent that learns from the reward signal alone, without surrendering the convergence, before-convergence, robustness, and auditability guarantees a high-stakes setting requires.
New Knowledge	New Knowledge 3b. A method for sourcing a reinforcement-learning agent's satisficing aspiration from a structured, dynamically updated Knowledge Graph, the Compiled-Aspiration MDP and the CA-RS-Q algorithm which keeps value learning and aspiration formation in separate channels, carries convergence, finite-time satisficing-rate, and graph-perturbation robustness guarantees, and renders each decision auditable to the facts that set the aspiration, for autonomous strategic execution in non-stationary E-CPSS.

7.1 Frame of Reference

Decision-making systems increasingly operate in worlds where the rules themselves shift mid-operation. A regulator changes a penalty schedule and the cost of inaction triples. A supplier disrupts and the cost of delay quintuples. A market regime turns, a government announces a tariff, a counterparty fails. The goals that were appropriate at the start of an operating period are no longer appropriate by its end. The decision-maker has to respond now, with the information at hand,

before the next decision arrives. There is no opportunity to pause the world, re-collect data, and retrain the underlying model from scratch.

This condition is the rule rather than the exception in the systems engineers increasingly take responsibility for. Supply chains, financial-instrument workflows, healthcare-resource allocation, energy-grid balancing, and regulated production schedules all share the structural property of exposing the agent to ongoing structural change. The change is not noise we can average out. It is signal we must act on. The standard reinforcement-learning toolkit was developed under different operating assumptions: a fixed Markov decision process, a stationary reward function, an environment that communicates only through reward signals. When the operating environment violates those assumptions, the gap between framework and reality becomes a source of decisions that are at best slow to adapt and at worst structurally incorrect.

What has changed alongside the worlds we deploy into is the form in which information about change arrives. In modern operating environments we no longer receive world-changes as raw signal we must infer from the reward stream. We receive them as named, time-stamped, structured facts: a regulator's penalty change is a recorded event with effective dates and scope; a supplier disruption is an entity-level fact reachable through a supply-chain graph; a policy update is a tagged announcement propagated through information systems. Enterprises now maintain knowledge graphs precisely to make this structure first-class, keeping entities, relations, and events available as queryable objects rather than as latent variables to be learned.

This shift has a name in the reinforcement-learning literature. Researchers formalize settings in which side-information determines the dynamics or rewards facing the agent as contextual Markov decision processes [1], and researchers in that lineage have continued to develop frameworks for environments whose context evolves over time. Through the contextual framing researchers accept that side-information is part of the operating environment and ask how to use it. The standard answer is that the context modifies the reward function or the transition dynamics of the underlying MDP, and the agent learns a context-aware policy. This is the right machinery for many problems. It is not the right machinery for the question we addressing this chapter.

The question we address in this chapter is what the agent should treat as good enough when the world has structurally changed. Consider an inventory operator at the moment a regulator raises the penalty for unmet demand. The MDP's reward function has not changed in the contextual sense; the agent still receives the same per-period reward signal from the environment. What has

changed is the threshold below which the operator would be unhappy with the policy's performance. An agent that uses the structured information to update its sense of what constitutes acceptable performance, and selects actions accordingly, is taking the structural change into account at the right level. An agent that does not is either waiting for the reward signal to accumulate enough evidence of the change to update its Q-table, which is slow, or refusing to act differently until a human operator overrides it, which is the failure mode we cannot tolerate in a regulated environment.

Through a principled mechanism we must do four things at once. We must preserve the convergence guarantees that make Q-learning trustworthy in the first place. We must provide a quantitative guarantee about the decisions we take under the structured information, not just an asymptotic claim. We must remain robust to errors in the structured information itself, since real knowledge graphs are noisy and incomplete. And we must produce decisions a reviewer can audit down to the specific facts that drove them, because "the model said so" is not a defensible answer in a regulated environment. None of the four is optional. If we satisfy only three of the four, the fourth remains a deployment-blocking liability.

We bring Simon's satisficing principle [2, 3] into reinforcement learning by sourcing the aspiration threshold from a structured knowledge graph. The threshold evolves automatically as the world changes, with provable convergence, satisficing rate, and graph-perturbation robustness guarantees, and every decision is fully auditable down to the named entities and edges that produced the threshold.

We use a mechanism with three structural elements. A compiler maps the active facts in the knowledge graph to a state-dependent aspiration $\rho(s)$. Designers program the agent to maintain a standard Q-table through Watkins-Dayan Q-learning [4] and to use the reward that the designers specify for the environment as the sole input to each update. We govern action selection through a risk-sensitive satisficing rule [7]. When a new fact arrives, we call the compiler again on the updated graph and obtain a new ρ; the Q-table is unchanged. When the new fact turns out to be wrong, only the action-selection criterion is affected; the learned Q-values are not contaminated. This separation between what the agent learns and what it aspires to is what allows convergence, satisficing-rate, and robustness guarantees to coexist in a single framework.

Researchers have previously used the term "aspiration-satisfying reinforcement learning" for a continuous-time multi-objective control framework with Hamilton-Jacobi-Bellman inequalities

[5]. We use the term in the discrete-time satisficing-RL sense and address a different research question; we discuss the relationship in Section 7.2.2. Researchers working on neurosymbolic reinforcement learning have independently used knowledge graphs as the substrate for encoding structured information that conditions decision-making [6]; their framework is complementary to ours, and we discuss it in Section 7.2.3.

Running example

Throughout the chapter, we ground the abstraction in a single running example: a multi-product inventory operation under news-driven structural shocks. The decision-maker chooses order quantities each week for one or two SKUs. The environment is a stochastic MDP with stockout penalties, holding costs, and bounded lead times. Through a pre-declared news schedule we introduce structured world-changes: supplier disruptions that raise effective stockout costs, regulator penalty changes that raise the cost ceiling, capacity announcements that relax it, and demand-regime shifts that alter the optimal order quantity. The knowledge graph contains the upstream entities (supplier, regulator) and the SKU-level facts that propagate from them. We use the example to ground the technical development, and none of our theorems depends on the inventory structure: Theorems 1 through 3 assume only a finite MDP, an admissible compiler, and the fact-set metric. The full setup is in Section 7.3.

Contributions

The contributions in this chapter are four claims, each tied to a specific later section.

First, we define the Compiled-Aspiration MDP (CA-MDP) and the CA-RS-Q algorithm (Section 7.4 and Section 7.5), and prove that under a settled knowledge graph the algorithm's Q-table converges to the optimal value function of the underlying MDP (Section 7.6, Theorem 1). We inherit the convergence property from Watkins-Dayan; our contribution is to show that it survives the compiled-aspiration mechanism unchanged.

Second, we prove a satisficing-rate guarantee: under a stationary knowledge graph and a mild gap condition, the action-selection rule produces actions satisfying the compiled aspiration on a non-negligible fraction of decisions (Section 7.6, Theorem 2). Through the result we generalize the bandit finite-regret guarantee of Tamatsukuri and Takahashi [7] to MDPs with state-dependent compiled aspirations.

Third, we prove a robustness theorem: the value of the policy under a perturbed knowledge graph differs from its value under the true knowledge graph by an amount bounded in the perturbation magnitude (Section 7.6, Theorem 3), and we provide a graph-distance-parameterized refinement (Section 7.6, Theorem 3-G). To our knowledge, this is the first result to bound reinforcement-learning *policy value* in terms of perturbations to an upstream symbolic structure that conditions action selection alone, leaving the agent's state observations, reward, and transition dynamics unchanged.

Fourth, we demonstrate structural auditability: every decision produced by the algorithm admits a derivation that traces the compiled aspiration back to a small set of named entities and edges in the knowledge graph (Section7.7). Auditability here is a property of the algorithm, not a post-hoc reconstruction.

Organization: In Section 7.2, we position the chapter against the satisficing reinforcement learning, contextual-MDP, knowledge-graph-in-RL, and interpretable-RL literatures. In Section 7.3, we specify the problem setting, the MDP machinery, and the inventory running example. In Section 7.4, we introduce the Compiled-Aspiration MDP formalism and the compiler axioms. In Section 7.5, we specify the CA-RS-Q algorithm. In Section 7.6, we prove the four theorems. In Section 7.7, we report empirical results on three benchmark scales (single-SKU, two-SKU, and three-SKU) and demonstrate the per-decision audit on a worked week. In Section 7.8, we discuss limitations and the relationship between our empirical findings and our theoretical contributions. In Section 7.9, we conclude. Readers interested only in the empirical results may proceed directly to Section 7.7; readers interested in the theoretical contributions should read Sections 3 through 6 in order.

7.2 Related Work

We position the present chapter against four research streams: satisficing and risk-sensitive reinforcement learning, the open question of where the aspiration level should come from, the integration of structured symbolic information into reinforcement learning, and the broader literature on interpretable and auditable decision-making. We close the section with a compact positioning summary in the form of a math-mapping table in which we separate what we borrow from prior work from what is new in the present chapter.

7.2.1 Satisficing and Risk-Sensitive Reinforcement Learning

The intellectual foundation of the present chapter is Herbert Simon's principle of satisficing, introduced as a model of bounded-rational decision making in human and organizational contexts [2, 3]. Simon's central claim was that real decision-makers operating under cognitive and informational constraints do not maximize a scalar utility function over the full action space; they search until an option meeting an aspiration threshold is found, and then they choose it. Researchers have applied the principle well beyond its original behavioral-economics setting, and researchers working on language-model alignment have re-derived satisficing as the framing for multi-faceted preference optimization at inference time [8].

The translation of satisficing to bounded rationality into reinforcement learning began with Takahashi and colleagues [9], who introduced the risk-sensitive satisficing (RS) approach. The structural form combines a reliability count τ, which grows with experience at an action, with a value-minus-aspiration term that compares the action's estimated value against a satisficing threshold. In the MDP setting addressed in the present chapter, we write this criterion as

$$\mathrm{RS}(s,a) = \tau(s,a) \cdot [Q(s,a) - \rho(s)], \tag{7.1}$$

where $\tau(s,a)$ is the reliability count at state-action pair (s,a), $Q(s,a)$ is the standard action-value estimate, and $\rho(s)$ is a state-dependent aspiration level. The RS rule selects actions that maximize this product, which has the effect of preferring actions whose value exceeds the aspiration with sufficient experience. Tamatsukuri and Takahashi [7] subsequently proved, in the K-armed bandit setting, that the RS rule converges to a satisficing arm and incurs finite regret when the aspiration is set so that exactly one arm is satisficing (that is, between the second-best and best mean rewards). We build on the bandit result in Theorem 2, where we generalize the finite-regret guarantee to Markov decision processes with state-dependent compiled aspirations.

Researchers have developed several parallel lines in the satisficing reinforcement learning literature. Reverdy, Srivastava, and Leonard [10] formulated satisficing objectives for bandits with Gaussian rewards, using a credible-limit framework distinct from RS. Michel and colleagues subsequently established regret bounds for satisficing in the multi-armed bandit [11], and Hajiabolhassan and Ortner [12] extended the analysis to satisficing in Markov decision processes under the average-reward criterion. The latter is the closest theoretical result to our Theorem 2:

their satisfaction level σ is a fixed scalar given to the learner, and they prove that the SAT-RL algorithm achieves expected regret bounded by a constant when σ lies between the optimal average reward and a known suboptimality gap. The setting differs from ours in two respects. First, our aspiration $\rho(s)$ is state-dependent and varies with the knowledge graph rather than being a single scalar provided exogenously. Second, our criterion is discounted-reward in finite-horizon episodes, while theirs is the infinite-horizon average-reward criterion. Both differences affect the proof structure, although the conceptual ground (satisficing rather than optimizing in MDPs) is shared.

Researchers in the risk-sensitive reinforcement learning literature use "risk-sensitive" in a sense distinct from the Takahashi RS sense. In the modern usage, risk-sensitive RL refers to optimizing a risk measure (CVaR, entropic risk, spectral risk) rather than expected return [13]. Our work draws on the satisficing lineage [7, 9], not the risk-measure lineage. The two share the goal of avoiding catastrophic outcomes but operationalize it through different mathematical objects: a satisficing threshold in our case, a risk measure in the modern risk-sensitive case.

Researchers have recently extended RS-style satisficing to function approximation and deep reinforcement learning. Minami, Kono, and Takahashi [14] introduced Linear RS (LinRS), a linear-function-approximation extension of the original tabular RS rule for contextual bandits, and Arumugam, Kumar, Gummadi, and Van Roy [15] developed a satisficing exploration procedure for deep reinforcement learning that sources aspirations from epistemic uncertainty over the optimal value function. The present chapter is orthogonal to both extensions: we retain the tabular structure to enable the theoretical guarantees, and we source aspirations from a structured knowledge graph rather than from a function approximator or an uncertainty estimate. We identify the function-approximation extension of CA-RS-Q as future work in Section 7.8.5.

7.2.2 Where the aspiration comes from

In the state of the art we see open question of how the aspiration ρ should be set. Tamatsukuri and Takahashi [7] treated ρ as a fixed scalar hyperparameter, with the regret guarantee conditional on ρ lying between certain bounds. Reverdy et al. [10] similarly treated the satisficing threshold as exogenous. Russo and Van Roy [16] proposed satisficing Thompson sampling for time-sensitive bandits, in which the learner targets a near-optimal rather than optimal action; they connect the choice of satisficing action to rate-distortion theory, and in their regret guarantee they assume the

learner knows the gap between the threshold and the optimal reward. Uragami, Sonota, and Takahashi [17] explicitly identified the determination of the aspiration level as an open problem and proposed a multi-agent solution in which agents share information about realized returns and update their aspirations from the shared signal.

The most direct precedent for our framing is the work of Mazouchi, Yang, and Modares [5] on aspiration-satisfying reinforcement learning. They study continuous-time nonlinear control with multiple objectives and use Hamiltonian inequalities to encode aspirations as constraints on Pareto-feasible policies, solved via sum-of-squares programming. Their terminology ("aspiration-satisfying RL") is the closest in the literature to ours, and their framework is the closest contemporary precedent for what we develop here. The frameworks differ in four substantive ways. First, they work in continuous-time optimal control with Hamilton-Jacobi-Bellman inequalities; we work in discrete-time Markov decision processes with the Watkins-Dayan Q-learning update. Second, they specify their aspirations externally, one per objective, and hold them fixed during learning; we compile ours from a knowledge graph and let them evolve as facts arrive and expire. Third, their solution procedure is iterative sum-of-squares programming over polynomial value-function representations; ours is standard temporal-difference learning with an aspiration-aware action-selection rule. Fourth, through their multi-objective framing they produce Pareto-optimal solutions to a constrained multi-objective problem; through our single-objective framing we produce a satisficing policy with bounded suboptimality on the primary objective.

Other treatments of aspiration adaptation include equilibrium-focused analyses in multi-agent stochastic games where ϵ-satisficing policy update rules guide independent learners to approximate equilibria [18]. Those authors work in a multi-agent, equilibrium-theoretic setting, and they do not address the single-agent question of where the threshold comes from. In this chapter, we contribute a third answer to the aspiration-source question, alongside the multi-agent broadcast of Uragami et al. [17] and the externally-specified objectives of Mazouchi et al. [5]: we propose that the aspiration be compiled from a structured knowledge graph by an axiomatically specified compiler.

7.2.3 Knowledge Graphs and Structured Information In Reinforcement Learning

Researchers combining structured symbolic information with reinforcement learning have matured the area into a substantial subfield. We organize the relevant work by the role the structured information plays in the algorithm.

Contextual and history-dependent MDPs. Through the contextual Markov decision process framework [1] researchers augment an MDP with a context variable that determines the reward function and transition dynamics, and the agent's task is to optimize across a distribution of contexts. Doshi-Velez and Konidaris [19] introduced hidden-parameter Markov decision processes for related settings in which the parameter is latent and must be inferred. Tennenholtz, Merlis, Shani, Mladenov, and Boutilier [20] generalized contextual MDPs to dynamic contexts that change over time non-Markovianly, introducing the DCMDP framework with upper-confidence-bound regret bounds. Our compiled-aspiration MDP (CA-MDP) is a descendant of the contextual MDP line: the fact set F_t is conceptually a context variable that varies with time. The principal distinction is that the context in CMDP and DCMDP modifies the reward and transition functions of the underlying MDP, while the fact set in CA-MDP modifies only the aspiration ρ, leaving the reward and transitions untouched. This is what allows CA-RS-Q to preserve the Watkins-Dayan convergence machinery (our Theorem 1) while inheriting the satisficing-regret machinery (our Theorem 2).

Knowledge graphs conditioning policy or action selection. Khandelwal, Yip, and Sheth [6] recently proposed a neurosymbolic reinforcement learning framework in which an editable knowledge graph encodes three categories of information: applicability rules capturing action preconditions and high-level effects, hard constraints defining action feasibility, and soft preferences shaping tradeoffs among feasible behaviors. Requirement changes become graph edits rather than policy rewrites, which they operationalize through constraint-based action shielding and preference-driven objective shaping. Their position paper is the closest contemporary work to ours in spirit, and it differs in mechanism: their knowledge graph drives action shielding and objective shaping, while ours sources an aspiration that drives satisficing action selection. Han, Dastani, and Wang [21] proposed neuro-symbolic action masking, in which a symbolic model learned in parallel with the policy produces action masks that rule out infeasible actions. Their

mechanism is again action shielding rather than aspiration sourcing. Researchers in a related line treat the knowledge graph as the environment of the reinforcement learning task itself, with graph navigation as the objective; the DeepPath algorithm of Xiong, Hoang, and Wang [22] is the canonical example, framing multi-hop relational reasoning over a knowledge graph as a policy-learning problem with continuous states based on graph embeddings. Our use of the knowledge graph as the source of a compiled aspiration is structurally distinct from all three uses.

Knowledge-graph-structured world models. Researchers in a separate line use knowledge graphs as the world model itself rather than as side information. Zhu, Li, and Elhoseiny [23] introduced the Value Memory Graph, in which they abstract the MDP as a directed graph of states and graph-actions for offline reinforcement learning. The graph here is the environment representation, not an auxiliary information source. Our knowledge graph is auxiliary throughout: the MDP retains its standard state and action structure, and the graph informs only the aspiration.

7.2.4 Reward shaping and the distinction from compiled aspiration

A natural question is whether the compiled aspiration $\rho(s)$ is equivalent to a potential function in the sense of Ng, Harada, and Russell [24], which would make CA-RS-Q a special case of potential-based reward shaping. It is not, and the distinction is structural rather than incidental.

In potential-based reward shaping we define an auxiliary reward

$$F(s, a, s') = \gamma\Phi(s') - \Phi(s) \tag{7.2}$$

that is added to the original reward $r(s, a, s')$ before the Bellman update. Ng et al. [24] proved that this transformation preserves the optimal policy of the underlying MDP. The mechanism is to modify the reward signal so that the agent learns the same optimal policy faster through denser feedback. The compiled aspiration in CA-RS-Q does not modify the reward signal. The Bellman update in CA-RS-Q is the same as in Q-learning,

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha\left[r + \gamma \max_{a'} Q(s', a')\right], \tag{7.3}$$

and the reward r is the unmodified environment reward. The aspiration $\rho(s)$ enters only the action-selection rule, through the RS value $\tau(s, a) \cdot [Q(s, a) - \rho(s)]$, not the update of Q. Consequently, the optimal Q^* that the algorithm converges to is the unmodified optimal value function of the

MDP, and the role of $\rho(s)$ is to bias action selection during finite-time exploration, not to alter the asymptotic target.

The distinction has empirical and theoretical consequences. Empirically, a misspecified ρ in CA-RS-Q causes the agent to select suboptimal actions during learning but does not corrupt the Q-table. In potential-based reward shaping, by contrast, the potential Φ is absorbed into the learned Q, as Ng et al. [24] showed by deriving $\tilde{Q}^*(s,a) = Q^*(s,a) - \Phi(s)$ for the shaped problem. A misspecified Φ that nonetheless satisfies the potential-based form preserves the optimal policy but can distort the magnitudes of the learned values; an additive reward modification that is not of potential-based form can change the optimal policy itself and is a known source of reward hacking [25]. In Theorem 1 we carry the Watkins-Dayan convergence result over unchanged. For reward shaping, by contrast, a designer must invoke the policy-invariance argument of Ng and co-authors [24], and that argument turns on whether the shaping term takes the potential-based form. The two mechanisms are not interchangeable, and CA-RS-Q is not a special case of reward shaping.

7.2.5 Interpretable and Auditable Reinforcement Learning

Researchers in the broader interpretable machine learning literature distinguish interpretability, a property of a model, from explainability, a post-hoc operation on a trained model. Lipton [26] argued that the term "interpretability" is overloaded and proposed a taxonomy that separates transparency from post-hoc explanation, with transparency itself admitting three sub-levels: simulatability (whether a person can contemplate the entire model at once), decomposability (whether individual components admit intuitive explanations), and algorithmic transparency (whether a person can understand the training procedure itself). Doshi-Velez and Kim [27] made a parallel call for a rigorous science of interpretable machine learning in which we define interpretability relative to a downstream task. Glanois and co-authors [28] survey the reinforcement learning specialization of these ideas, distinguishing interpretable inputs, interpretable transition and reward models, and interpretable decision-making. Vouros [29] provides a complementary survey focused on deep reinforcement learning, with emphasis on the audit and trust requirements that arise in high-stakes deployment.

The auditability property we establish for CA-RS-Q in Section 7.7.6 sits in the "interpretable decision-making" branch of the Glanois et al. [28] taxonomy: the algorithm produces a per-

decision derivation that traces every contribution to a specific named entity in the knowledge graph and a specific edge along which the contribution travels. The contrast with post-hoc explanation methods is structural. Methods that produce explanations after the fact (saliency maps, counterfactual generation, surrogate decision trees) operate on a trained black-box policy and produce approximations of the policy's reasoning that may or may not be faithful to the underlying decision [30]. CA-RS-Q's audit trail is the underlying decision, in the same sense that a derivation in mathematics is the proof rather than an approximation of it. The auditability is a property of the algorithm, not a post-hoc reconstruction.

7.2.6 Positioning Summary: What We Borrow and What is New

We close this section with an explicit separation of borrowed components from new contributions, organized so that the contribution is straightforward to verify.

Table 7.1: Borrowed versus new components in the present chapter

Component	Source	Role in our chapter
Borrowed		
Q-learning update rule	Watkins & Dayan [4]	Standard temporal-difference learning of Q; unchanged in CA-RS-Q
RS value function $\tau \cdot [Q - \rho]$	Takahashi et al. [9]	Action-selection criterion; reused with state-dependent ρ
Reliability count $\tau(s, a)$ update	Takahashi et al. [9]	Confidence weight in the action-selection criterion
Bandit finite-regret proof template	Tamatsukuri & Takahashi [7]	Template generalized in our Theorem 2 from bandits to MDPs
Stochastic approximation machinery	Robbins & Monro [31]; Bertsekas & Tsitsiklis [32]; Borkar [33]	Convergence proof of our Theorem 1
MDP and Q-learning convergence machinery	Puterman [34]; Watkins & Dayan	Theoretical foundation for Theorem 1

Component	Source	Role in our chapter
	[4]; Sutton & Barto [35]	
New (this chapter)		
Knowledge graph F_t as a formal state variable alongside the MDP state	New	First integration of a KG as a separate state variable in a satisficing RL framework
Compiler axioms (Lipschitz continuity, monotonicity, identity at empty KG)	New	Axiomatic specification of what a fact-to-aspiration function must satisfy (Section 7.4.2)
Compiled-Aspiration MDP (CA-MDP) formalism	New	Distinct from CMDP [1], DCMDP [20], HiP-MDP [19]
Compiler operator $\mathcal{C}: 2^F \times S \to \mathbb{R}$	New	New mathematical object; concrete instance given in Section 7.4.4
CA-RS-Q algorithm	New	Combination of RS action selection with compiled ρ from the KG
Theorem 1 (convergence under settled KG)	New	Extends Watkins-Dayan to the compiled-aspiration setting
Theorem 2 (satisficing rate under stationary KG)	New	Extends Tamatsukuri-Takahashi finite-regret to MDPs with compiled, state-dependent aspirations
Theorem 3 (KG-robustness)	New	First bound on RL policy value under perturbation of an upstream symbolic structure that conditions action selection (distinct from perturbations of state observations, reward, or transitions)

Component	Source	Role in our chapter
Theorem 3-G (graph-structured robustness)	New	Lipschitz bound parameterized by graph edit distance
Per-decision audit trail derived from the compiler	New	Structural auditability distinct from post-hoc explanation methods (Section 7.2.5, Section 7.7.7)

The contributions in the lower half of the table together constitute the new knowledge content of the chapter. The borrowed components in the upper half are cited at the points of use throughout Sections 5 and 6.

7.3. Problem Setting

We work in the setting of a decision-maker who operates a recurring process and has access to a structured representation of the world relevant to that process. We are interested in how the decision-maker should respond to changes in the structured representation, with formal guarantees on the response and full auditability of every recommendation.

In this section, we introduce the three components we work with. The first is the recurring decision itself, modeled as a Markov decision process. The second is a concrete instance of such a process drawn from inventory management, which we carry as a running example throughout the chapter. The third is the knowledge graph, which we treat as a first-class object alongside the Markov decision process.

7.3.1 Markov Decision Processes

A finite Markov decision process is the tuple

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma) \tag{7.4}$$

where S is a finite set of states, A is a finite set of actions, P(s' | s, a) is the transition probability, R(s, a) is the reward function, and $\gamma \in [0, 1)$ is the discount factor. For a policy π that maps states to actions, the action-value function is

$$Q^{\pi}(s,a) \;=\; \mathbb{E}^{\pi}\left[\sum_{k=0}^{\infty} \gamma^{k}\; R(s_{t+k}, a_{t+k}) \,|\, s_t = s,\; a_t = a\right] (7.5)$$

and the optimal action-value function is $Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a)$. We denote by $R_{\max}$ the maximum absolute reward, $R_{\max} = \max_{s,a}|R(s,a)|$. We follow standard references for background [34, 35].

7.3.2 The Inventory Management Running Example

We ground the formal development in a concrete example we carry throughout the chapter. We consider a firm that orders a single product, which we name SKU_H, from a single supplier, which we name Supplier_X. The firm makes one ordering decision per week and operates on a 52-week horizon.

At the start of week t the firm observes its on-hand inventory I_t and its outstanding backlog B_t. We treat I_t and B_t as integer-valued with caps $I_{\max}$ and $B_{\max}$. The firm chooses an order quantity o_t from a finite set of allowed quantities. We model the supplier as delivering a stochastic fraction of o_t during the week. We model customer demand D_t as Poisson with rate λ. At the end of the week the firm serves as much demand as its available stock permits; we treat the unserved demand as new backlog and cap the remaining inventory at $I_{\max}$.

The firm incurs three costs per week. An order cost c_{order} per unit ordered, a holding cost c_{hold} per unit of inventory remaining at the end of the week, and a backlog cost c_{backlog} per unit of outstanding backlog. We define the reward at each step as the negation of the total cost incurred that week.

This setup is a small finite Markov decision process. For the parameter values we use in our experiments, the state space has 66 elements and the action space has 5 elements. We give the full specifications in Section 7.7.1. We compute the optimal policy by value iteration and use it as a ground-truth reference in our convergence experiments.

We refer to this single-product version as Benchmark A. We also use a two-product extension we call Benchmark B in Section 7.7, with 4,356 states and 16 actions, to verify that our results transfer to a problem at a comparatively higher scale. We use the same fact schema and the same compiler form for both benchmarks (the per-fact-contribution structure of Section 7.4.4), with benchmark-specific per-type coefficients and baseline; the state and action spaces, the per-type

coefficients, and the baseline differ across the two benchmarks while the framework's compiler structure is the same.

7.3.3 Knowledge Graph and Facts

A knowledge graph in our setting is the triple

$$G = (V, E, F)$$

where V is a finite set of typed entity nodes, E ⊆ V × V × Σ is a finite set of typed directed edges, and F is a finite set of facts referenced to entities. In the inventory example, V contains two products we name SKU_H and SKU_M, a supplier we name Supplier_X, a regulator we name Regulator_A, and a region we name Region_North. The edge set E contains four static relationships: (Supplier_X, SKU_H, supplies), (Supplier_X, SKU_M, supplies), (Supplier_X, Region_North, located_in), and (Regulator_A, Region_North, regulates). These relationships hold throughout the episode unless we explicitly add new edges as part of fact arrival. We use this mechanism for one fact type in our benchmark schema: when a regulator_penalty_change fact arrives carrying an entity reference to a regulator node, the compiler adds a regulates edge from the regulator to each product entity in the regulator's jurisdiction at the time of arrival, and that edge persists in the graph for as long as the fact remains in its validity window. The supplier-related fact types (supplier_disruption, capacity_increase) attach to nodes already connected to product entities by the static supplies edges and do not extend the edge set.

A **fact** is the tuple

$$f = \left(id,\ type,\ \sigma,\ v,\ t_0,\ t_1,\ c,\ \tau_f\right) \tag{7.6}$$

where id is a unique identifier, $type$ is drawn from a finite set of fact types Θ, σ specifies the scope of the fact (which entities the fact applies to; the symbol σ here is the fact scope and is distinct from the edge-label alphabet Σ used in $E \subseteq V \times V \times \Sigma$), $v \in \mathbb{R}$ is a numeric severity reading with $|v| \geq v_{\min} > 0$ for some fixed lower bound $v_{\min}$, $[t_0, t_1]$ is the validity window in discrete time, $c \in (0,1]$ is a confidence value, and $\tau_f \in \{+1, -1\}$ is a tag distinguishing facts that we treat as tightening from facts that we treat as loosening. (The symbol τ_f here is the fact tag; the reliability count $\tau(s,a)$ introduced in Section 7.5.1 is a distinct quantity sharing the symbol by convention; they do not interact.) The lower bound $v_{\min} > 0$ on severity magnitude is a schema-

level admissibility requirement; in the benchmark implementation it is enforced by the news generator (the per-fact-type severity-sampling range has lower endpoint $\geq v_{\min}$ by construction), and conflict resolution preserves it because the resolved severity magnitude is the maximum of the input magnitudes.

In the benchmark implementation, the scope σ is encoded as a key-value pair selecting one entity (typically an SKU); the formal model accommodates richer scope structures, but in the benchmarks reported we here use the SKU-level encoding. Examples in the inventory setting include a supplier disruption referenced to Supplier_X with severity 0.7 and tag +1, a demand surge referenced to SKU_H with severity 1.4 and tag +1, and a capacity increase referenced to Supplier_X with severity 0.15 and tag −1. The first two communicate that conditions are more adverse than baseline. The third communicates that conditions are more favorable than baseline.

The **active fact set at time t** is

$$F_t \;=\; \{\, f \in \mathcal{F}_{\text{total}} \text{ received by time } t : t_0(f) \leq t \leq t_1(f) \,\} \tag{7.7}$$

where $\mathcal{F}_{\text{total}}$ denotes the cumulative set of facts received by time t. We treat a fact as no longer active once t exceeds $t_1(f)$. We assume $|F_t|$ is finite for all t and that the arrival process for new facts is exogenous to the decisions taken by the learner. We further assume the active fact set contains at most one fact per (type, σ) pair: when duplicate arrivals occur, the implementation collapses them via max-severity, max-confidence resolution before compiler evaluation, and in the formal analysis we treatsthis resolved set as F_t throughout.

We define a distance between fact sets that we will need in Section 7.6. For $F, F' \subseteq \mathcal{F}$,

$$d_F(F, F') = \sum_{f \in F \triangle F'} c_f \cdot |v(f)| \tag{7.8}$$

where F △ F' = (F \ F') ∪ (F' \ F) is the symmetric difference. We verified that d_F is non-negative, satisfies identity, is symmetric, and respects the triangle inequality, so it is a metric on fact sets.

7.3.4 The Compiler and the Aspiration

The third component of our framework is a function we call the **compiler**. The compiler takes a state and a fact set and returns a real number we call the aspiration. We write

$$\rho_t(s) = C(s, F_t) \tag{7.9}$$

for the aspiration at state s and time t. We use the convention that the satisficing condition is $Q(s,a) \geq \rho$. A smaller (more negative) value of ρ admits more actions and is therefore **looser**; a larger (less negative) value of ρ admits fewer actions and is **stricter**. The intuition is the cost-ceiling reading: in our cost-minimization MDPs, ρ is negative, and we interpret the absolute value of ρ as the maximum acceptable long-run cost the agent is targeting (the cost ceiling). A higher cost ceiling corresponds to a more negative ρ and admits more actions, hence looser; a lower cost ceiling corresponds to a less negative ρ and admits fewer, hence stricter. Under this convention, **tightening facts** ($\tau_f = +1$) signal adverse external conditions and push ρ downward, which raises the cost ceiling and loosens the satisficing threshold (the agent accepts higher-cost actions reflecting the harder external conditions); **loosening facts** ($\tau_f = -1$) signal favorable conditions and push ρ upward, which lowers the cost ceiling and tightens the threshold.

We treat the compiler as a fixed and known function in this chapter. We do not make any assumption about how the compiler is obtained beyond a small number of axioms that we state in Section 7.4. The compiler we use in our experiments is a sum of per-fact contributions, with each contribution being a confidence-weighted function of severity. The axioms we impose are consistent with many alternative compiler designs.

7.3.5 Putting the Pieces Together

We now have three objects. The first is a Markov decision process that models the recurring decision. The second is a knowledge graph whose contents we update over time as new facts arrive and old facts pass out of their validity windows. The third is a compiler we define to map the current knowledge graph and the current state to a real-valued target.

The Markov decision process is standard. The knowledge graph and the compiler together provide what we call the compiled aspiration. In Section 7.4 we combine these three objects into a formal extension of the Markov decision process and state the axioms we require of any admissible compiler. In Section 7.5 we describe the learning algorithm we design to pursue the compiled aspiration. In Section 7.6 we prove four theorems about the resulting construction, covering convergence, satisficing rate, and robustness to changes in the knowledge graph.

7.4. The Compiled-Aspiration MDP

In Section 7.3 we introduced three objects separately: a Markov decision process, a knowledge graph, and a compiler. In this section we combine them into a single formal object we call the Compiled-Aspiration MDP. We then state the four axioms we require of any admissible compiler. We extend one of the axioms to the graph-structured setting. We close with the concrete compiler we use in our experiments and a worked numerical example.

7.4.1 The Compiled-Aspiration MDP

We define a **Compiled-Aspiration Markov Decision Process** (CA-MDP) as the tuple

$$\mathcal{M}_{\mathrm{CA}} = (\mathcal{S}, \mathcal{A}, P, R, \gamma, \mathcal{F}, P_F, C) \tag{7.10}$$

where $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ is a standard finite Markov decision process as in Section 7.3.1, $\mathcal{F}$ is the set of admissible facts as in Section 7.3.3, P_F is the exogenous arrival process for facts (formally, a discrete-time stochastic process on the fact universe $\mathcal{F}$, with the per-step arrival count almost surely finite — sufficient for $|F_t| < \infty$ to hold for all t), and C is the compiler.

The observable state at time t is the pair (s_t, F_t), where s_t is the standard MDP state and F_t is the active fact set. The aspiration at state s and time t is

$$\rho_t(s) = C(s, F_t) \tag{7.11}$$

We make a deliberate design choice at this point. The aspiration ρ_t(s) appears in the action-selection rule of our learning algorithm in Section 7.5, but it does not appear in the Q-learning update. This means that as F_t changes, the aspiration changes, and the actions the agent takes under our policy change, while the value estimates continue to track the underlying optimal Q-function as if no fact had arrived. We return to this design choice in Section 7.5 and motivate it in detail. It is the property under which Theorem 1 reduces to the classical Q-learning convergence result.

In a constrained MDP we impose constraint functions on policy values, not an aspiration on per-state action selection. In a dynamic contextual MDP [20] we augment the state with a general context, without the compiler structure or the satisficing semantics. In a standard contextual MDP we condition the dynamics on context, not the aspiration. We are not aware of a prior formalism that augments an MDP with a fact set and a compiler that conditions the aspiration alone, leaving reward and transitions unchanged.

7.4.2 Axioms of an Admissible Compiler

A compiler is any function

$$C: \mathcal{S} \times 2^{\mathcal{F}} \to \mathbb{R} \tag{7.12}$$

.

We do not assume any particular functional form. Instead, we require any compiler we are willing to use to satisfy four axioms.

Axiom A1 (Empty-graph identity). There exists a baseline function ρ_0: S → $\mathbb{R}$ such that

$$C(s, \emptyset) = \rho_0(s) \quad \text{for all } s \in \mathcal{S} \tag{7.13}$$

When no facts are active, the compiler reduces to a fixed baseline determined by the state alone. We obtain ρ_0(s) in our experiments by policy evaluation against a designer-specified baseline policy.

Axiom A2 (Lipschitz continuity in fact-set distance). There exists a constant $L_C > 0$ such that for all states s and all fact sets $F, F' \subseteq \mathcal{F}$,

$$|C(s, F) - C(s, F')| \;\leq\; L_C \cdot d_F(F, F') \tag{7.14}$$

Small changes in the active fact set produce small changes in the aspiration. We define metric d_F sin Section 7.3.3.

Axiom A3 (Monotonicity under tagged inclusion). For F ⊆ F':

- If every f ∈ F' \ F has tag τ_f = +1, then C(s, F') ≤ C(s, F) for all s.
- If every f ∈ F' \ F has tag τ_f = −1, then C(s, F') ≥ C(s, F) for all s.

When we add only tightening facts to the active set, the aspiration is lower, which raises the cost ceiling and loosens the target; when we add only loosening facts, the aspiration is higher, which lowers the cost ceiling and tightens it. Through this axiom we encode the tag semantics from Section 7.3.3.

Axiom A4 (Separability). There exist a baseline function $\rho_0: \mathcal{S} \to \mathbb{R}$ and a contribution function $\Phi: 2^{\mathcal{F}} \times \mathcal{S} \to \mathbb{R}$ such that

$$C(s,F) = \rho_0(s) + \Phi(F,s) \tag{7.15}$$

with $\Phi(\emptyset, s) = 0$ for all s, and with Φ Lipschitz continuous in F under d_F with some constant $L_C > 0$. By setting $F = \emptyset$ we recover the (A1) baseline as $\rho_0(s) = C(s, \emptyset)$, and the Lipschitz constant of Φ is the L_C we name in (A2); (A4) then, we introduce both quantities rather than presuppose them.

Under separability, we can decompose the compiler output into a state-dependent baseline and a fact-dependent contribution. We use this decomposition to construct the audit trail in Section 7.7.6: we attribute every component of the aspiration either to the baseline or to a specific fact.

On axiom dependence. The four conditions (A1)–(A4) are not logically independent. (A1) follows from (A4) by setting $F = \emptyset$ and reading $\rho_0(s) := C(s, \emptyset)$ as a derived quantity, and (A2) follows from (A4)'s Lipschitz Φ. (A3) is logically independent of the other three and must be imposed separately. The admissibility of a compiler thus reduces, substantively, to (A3) and (A4); we state (A1) and (A2) as named conditions because Section 7.6's theorems invoke them by name. We denote the class of admissible compilers by $\mathcal{C}$.

7.4.3 The Graph-Structured Extension

We now extend the axiom system to the graph-structured knowledge graph defined in Section 7.3.3. We have the knowledge graph as G = (V, E, F). When the knowledge graph has nontrivial entity and edge structure, we require the compiler to operate on a restricted fact set determined by the graph topology.

We define two entity-attachment functions used throughout this section. For a state $s \in \mathcal{S}$, the **state entity set** $\mathrm{ent}(s) \subseteq V$ is fixed by the benchmark specification. For Benchmark A of Section 7.3.2, $\mathrm{ent}\big((I, B)\big) = \{\text{SKU\_H}\}$. For Benchmark B (Section 7.7.1), $\mathrm{ent}\big((I_H, I_M, B_H, B_M)\big) = \{\text{SKU\_H,SKU\_M}\}$. For a fact $f \in \mathcal{F}$, the **fact entity set** $\mathrm{ent}(f) \subseteq V$ is read from the scope field σ: when σ specifies one or more entities, $\mathrm{ent}(f) := \sigma$ (entity-attached fact); when σ is empty, $\mathrm{ent}(f) := \emptyset$ (graph-wide fact). The graph-wide case is the convention by which we express facts that apply to the operating environment as a whole rather than to any specific entity.

For a state s with state entities ent(s) ⊆ V, the **1-hop neighborhood** of s in G is

$$N_1(s, G) \;=\; \text{ent}(s) \;\cup\; \{\, v' \in V : \exists\, v \in \text{ent}(s) \text{ such that } (v, v', \sigma) \in E \text{ or } (v', v, \sigma) \in E \,\} \tag{7.16}$$

The **state-relevant facts** are then

$$F_s(G) \;=\; \{\, f \in F : \text{ent}(f) \cap N_1(s, G) \neq \emptyset \,\} \;\cup\; \{\, f \in F : \text{ent}(f) = \emptyset \,\} \tag{7.17}$$

A fact is **state-relevant** at s only when its entity reference is either among the state's own entities or one edge away. We treat graph-wide facts (those with no entity references) as relevant at every state.

The compiler operating on the graph-structured knowledge graph is the same compiler as before, applied to the restricted fact set:

$$C(s, G) = C\big(s, F_s(G)\big) \tag{7.18}$$

We now extend Axiom A2 to graph distances. Recall the graph distance metric

$$d_G(G, G') \;=\; w_V \cdot |V \,\triangle\, V'| \;+\; w_E \cdot |E \,\triangle\, E'| \;+\; d_F(F, F') \tag{7.19}$$

defined here for use in (A2-G) below. We require $w_V \geq 1$ and $w_E \geq 1$ so that the Lipschitz constant L_G derived in Section 7.6.4 carries no weight-dependent factors. In all benchmarks reported here we take $w_V = w_E = 1$.

Axiom A2-G (Lipschitz continuity in graph distance). There exists a constant $L_G > 0$ such that for all states s and all knowledge graphs G, G',

$$|C(s, G) - C(s, G')| \;\leq\; L_G \cdot d_G(G, G') \tag{7.20}$$

We show in Section 7.6.4 that under A1, A2, A3, and A4 plus boundedness of the per-fact contribution, A2-G holds with

$$L_G \;=\; \max\{\, L_C,\; c_{\max} \cdot M_{\max} \cdot \varphi_{\max} \,\} \tag{7.21}$$

where $c_{\max}$ bounds the per-fact confidence, $M_{\max}$bounds the per-type contribution magnitude, and where $\varphi_{\max}$ denotes the maximum number of facts attached to any single entity in $F \cup F'$. A single new edge or node can pull at most $\varphi_{\max}$ facts into the 1-hop neighborhood of some state.

When V = V' = ∅ and E = E' = ∅, the graph distance reduces to the fact-set distance and L_G reduces to L_C, so A2-G specializes to A2 in the flat case.

7.4.4 The Concrete Compiler

We give an explicit compiler that satisfies all four axioms and that we use throughout the experiments. The construction is a sum of per-fact contributions:

$$C(s, F) \;=\; \rho_0(s) \;+\; \sum_{f \in F} c\,(f) \cdot \psi_{\text{type}(f)}(f, s) \tag{7.22}$$

Here $\rho_0 : \mathcal{S} \to \mathbb{R}$ is the baseline derived from (A4) at $F = \emptyset$ (equivalently, the baseline named in (A1)), and $\{\psi_\theta\}_{\theta \in \Theta}$ is a family of per-type contribution functions, with Θ denoting the finite set of fact types introduced in Section 7.3.3. Each ψ_θ maps a fact of type θ and a state to a real number. The signature admits state-dependent ψ_θ, but the per-type contribution functions used throughout the benchmarks of this chapter are state-independent (the worked examples of Section4.5 and Section7.6 reflect this), and our concrete compiler instantiates each ψ_θ as a function of the fact f alone. The state-dependent generality is preserved in the formal model for future use. The schema admits only configurations under which the per-type ψ functions preserve (P1)–(P3) for all admissible severity values; for the order_cap type in particular, this constrains the per-fact normalizer in the compiler config to lie at or above the maximum admissible severity in the schema, so that the linear form $\psi_{\text{order\_cap}}(f) = \text{coef} \cdot \left(1 - v_f/\text{normalizer}\right)$ with $\text{coef} < 0$ remains nonpositive for all v_f a tightening fact can take.

We require each ψθ to satisfy three properties:

- **(P1) Boundedness.** $|\psi_\theta(f, s)| \le M_\theta$ for all facts f of type θ and all states s
- **(P2) Sign alignment with the tag.** If $\tau_f = +1$ then $\psi_\theta(f, s) \le 0$; if $\tau_f = -1$ then $\psi_\theta(f, s) \ge 0$.

- **(P3) Lipschitz continuity in the severity reading.** For facts f, f' of the same type and same scope, $|\psi_\theta(f, s) - \psi_\theta(f', s)| \leq K_\theta \cdot |v(f) - v(f')|$.

We verify in Section 7.6 (as a lemma supporting Theorems 3 and 3-G) that this compiler satisfies A1, A2, A3, and A4 with $L_C \leq M_{\max}/v_{\min}$, where $M_{\max} = \max_\theta M_\theta$ and $v_{\min}$ is a lower bound on the severity magnitude of any admissible fact. The Lipschitz bound is not always tight.

The computational cost of one compiler call is O(|F|) in the flat case and O(|F_s(G)|) in the graph-structured case. Both are linear in the number of facts that reach the decision.

7.4.5 A Worked Example

Consider state $s = (I = 0, B = 3)$ at week $t = 39$ of an episode in which two facts are active: a regulator penalty change attached to Regulator_A with severity $v = 3.0$, confidence $c = 0.95$, and tag $\tau_f = +1$; and a capacity increase attached to Supplier_X with severity $v = 0.15$, confidence $c = 0.75$, and tag $\tau_f = -1$.

The state's entity set is $\text{ent}(s) = \{\text{SKU\_H}\}$ (per the Benchmark A specification of Section 7.3.2). When the regulator fact arrived, the compiler added a regulates edge from Regulator_A to SKU_H per the rule of Section 7.3.3. The 1-hop neighborhood is therefore $N_1(s, G) = \{\text{SKU\_H,Supplier\_X,Regulator\_A}\}$: Supplier_X is reached through the static supplies edge, and Regulator_A is reached through the regulates edge added at fact arrival. Both active facts are therefore in $F_s(G)$ via the entity-overlap clause.

The baseline aspiration for this state, obtained from policy evaluation under a designer-specified order-up-to baseline policy with target level $S = \text{round}(1.5\lambda)$ (yielding $S = 6$ at $\lambda = 4$ for Benchmark A, and $(S_H, S_M) = (6,9)$ at $(\lambda_H, \lambda_M) = (4,6)$ for Benchmark B), is $\rho_0(s) = -863$ cost units. The per-type contributions from the concrete compiler are

$$c \cdot \psi_{\text{regulator\_penalty\_change}} \;=\; 0.95 \cdot (-400) \;=\; -380$$

$$c \cdot \psi_{\text{capacity\_increase}} \;=\; 0.75 \cdot (+2.55) \;\approx\; +2$$

so the compiled aspiration is

$$\rho_t(s) \;=\; -863 + (-380) + 2 \;=\; -1241$$

.

Because the regulator penalty fact is active, the compiled aspiration falls by 380 cost units (a tightening contribution per the convention of Section 7.3.4, signaling adverse external conditions); the capacity increase raises it by approximately 2 cost units (a loosening contribution signaling favorable conditions). The net effect is a much lower aspiration than the baseline ($\rho = -1241$ versus $\rho_0 = -863$), which under the operational convention of Section 7.3.4 corresponds to a looser satisficing threshold: the agent now accepts as satisficing actions whose long-run cost would have been considered too high in the absence of the regulator fact, reflecting the elevated regulatory penalty environment.

We extend this kind of derivation in Section 7.7.7 to a multi-SKU Benchmark B episode to illustrate the algorithm's per-decision audit trail at richer scale.

Summary: We defined the Compiled-Aspiration MDP and stated four axioms an admissible compiler must satisfy. We extended one of these axioms to the graph-structured setting. We constructed an explicit compiler used in our experiments and verified its admissibility. In Section 7.5 we describe the learning algorithm that uses this construction.

7.5 The CA-RS-Q Algorithm

Through the Compiled-Aspiration MDP of Section 7.4 we specify what we want a learning agent to act on at each step: a state, an action set, and an aspiration we compute from the current knowledge graph. We now describe the learning algorithm we use to act on it. We call this algorithm Compiled-Aspiration Risk-Sensitive Q-Learning, abbreviated CA-RS-Q. In the algorithm we combine tabular Q-learning with the risk-sensitive satisficing value function of Takahashi and co-authors [9], modifying it so that the aspiration in the action-selection rule is the compiled aspiration ρ_t(s) rather than a fixed designer-supplied threshold. In the remainder of this section we specify the quantities we maintain in memory, the rule by which we select an action, the rule by which we update memory after we observe a reward, and the design rationale for the most important choice we make.

Before we specify CA-RS-Q formally, we give an overview in Figure 7.1, which we organize as

three panels ordered from intuition, to computation, to behavior over time.

In Panel 1, we present the mental model behind the algorithm. At each step, we compile the facts that are active in the knowledge graph into a single aspiration, ρ, the cost we are willing to treat as good enough under the current conditions, and we judge each candidate action by how far its value exceeds this aspiration, weighting each option by how often we have tried it. This is the satisficing stance we adopt throughout: we do not select the lowest-cost action, but we accept any action whose value meets the compiled aspiration, and among these we prefer well-tried actions to higher-valued but less-tried ones. We use the compiled aspiration to choose among actions, but not in how we learn an action's value, so we continue to learn the true costs of actions even as we revise the goal.

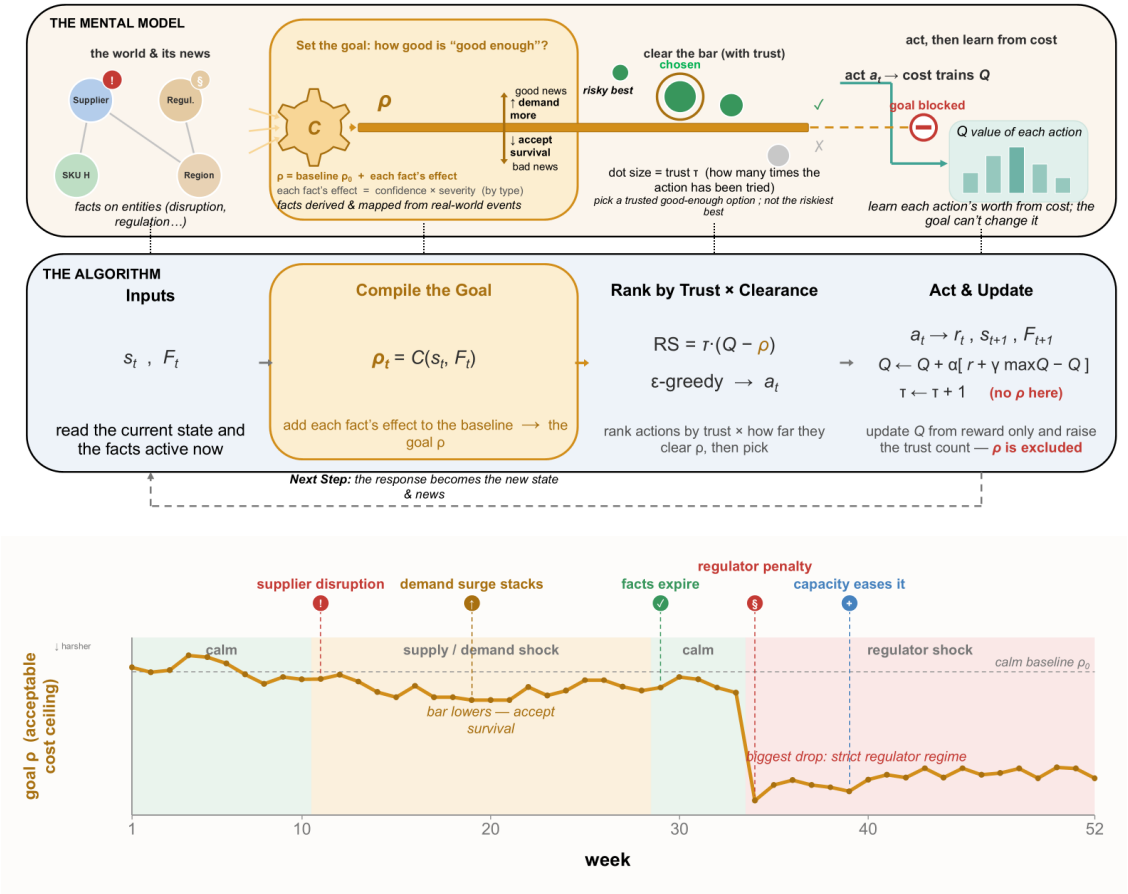


Figure 7.1: Overview of CA-RS-Q

In Panel 2, we restate the same step as a computation, and we preview the quantities that we define in the rest of this section. We maintain a value estimate $Q(s, a)$ and a trust count $\tau(s, a)$ for every

state–action pair. At each step we read the current state s_t and the active facts F_t; we compile the aspiration $\rho_t = C(s_t, F_t)$; we score the candidate actions by the risk-sensitive value $RS_t = \tau \cdot (Q - \rho_t)$; we select an action ε-greedily; and, after we observe the reward, we update Q by standard Q-learning and we increment τ. We align Panels 1 and 2 column by column, so that we can read each element of the mental model against the corresponding quantity in the computation. In Panel 3, we show why we compile the aspiration rather than fix it. We trace ρ over a representative horizon during which different facts are active at different times. When conditions are calm, ρ_t equals the baseline ρ_0. While a supplier disruption and then a demand surge are active, we sum their effects and compile a lower aspiration, and we accept higher operating cost to remain feasible. Once these facts are no longer active, we recover the baseline aspiration without further intervention. While a high-severity regulatory change is active, we compile a much lower aspiration, and we offset part of this reduction when a capacity increase is also active. In each case we re-aim the goal directly from the facts, step by step, and we do so without relearning the property that we design the update rule to protect.

We state the complete per-step procedure in Algorithm 7.1. We give it here, ahead of the detailed specification, so that we can refer to it through the rest of the section: in Section 7.5.1 we define the quantities we maintain in memory and how we initialize them; in Section 7.5.2 we define the compiled aspiration $\rho_t = C(s_t, F_t)$ and the risk-sensitive value $RS_t = \tau \cdot (Q - \rho_t)$ that we use within it; in Section 7.5.3 we specify the action-selection rule that uses them; in Section 7.5.4 we specify the update rule by which we revise Q after we observe a reward; and in Section 7.5.5 we give the rationale for the choice by which we distinguish CA-RS-Q from a conventional risk-sensitive learner that we use the compiled aspiration in action selection but exclude it from the value update.

Algorithm 7.1 CA-RS-Q (Compiled-Aspiration Risk-Sensitive Q-learning)

Require: exploration rate $\varepsilon \in (0, 1)$; discount $\gamma \in [0, 1)$; initial value Q_0; compiler C; finite action set A; learning-rate exponent $p \in (0.5, 1]$ (we use $p = 0.7$).

Initialize (once; persists across episodes): $Q(s, a) \leftarrow Q_0$ and $\tau(s, a) \leftarrow 0$ for all $(s, a) \in S \times A$.

for each environment step $t = 0, 1, 2, \ldots$ **do**

1: Observe the current state s_t and the current active fact set F_t.

2: Compile the aspiration $\rho_t \leftarrow C(s_t, F_t)$. ▷ *a single real number; recomputed every step, never cached*

3: **for** each $a \in A$ compute $\mathrm{RS}_t(s_t, a) \leftarrow \tau(s_t, a) \cdot [Q(s_t, a) - \rho_t]$.

4: Select the action a_t by ε-greedy on the RS values:

with probability ε: $a_t \leftarrow$ an action drawn uniformly at random from A;

otherwise: $a_t \leftarrow \arg\max_{a \in A} \mathrm{RS}_t(s_t, a)$ (ties broken uniformly at random).

Execute a_t in the environment; observe the reward r_t and the next state s_{t+1}.

5: Update the value estimate (standard one-step Q-learning; ρ_t does not appear):

$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \cdot [r_t + \gamma \cdot \max_{a' \in A} Q(s_{t+1}, a') - Q(s_t, a_t)]$,

with $\alpha_t = (1 + \tau(s_t, a_t))^{-\mathrm{p}}$, where $\tau(s_t, a_t)$ is the reliability count before its increment in step 6.

6: Increment the reliability count: $\tau(s_t, a_t) \leftarrow \tau(s_t, a_t) + 1$.

end for

Note. The aspiration ρ_t enters action selection (steps 3–4) but not the value update (step 5); routing ρ_t into selection only, with no path into the Q-update, is the design property used in Theorem 1 (Section6.1) and relied on throughout Section6.

We now specify each part of Algorithm 7.1 in turn, beginning with the quantities we hold in memory.

7.5.1 Quantities Maintained During Execution

At each time step we maintain two tables, each of the same shape as the Q-table of standard tabular Q-learning. The first is the action-value estimate Q(s, a), which we initialize to a fixed value Q0 and update after every observation in the standard Q-learning manner we specify in Section 7.5.4. The second is the reliability count τ(s, a), which we initialize to zero and increment by one each time we take action a from state s. Over the course of training, τ(s, a) records the number of times we have visited the pair (s, a), and serves as a measure of how much evidence we have accumulated about the value of taking a from s. The reliability count is the central novelty inherited from Takahashi et al.'s risk-sensitive satisficing framework. In standard Q-learning, every state-action pair is treated equivalently at action-selection time, and the action with the highest current Q estimate is chosen regardless of how trustworthy that estimate is. Through the reliability count we

weight an action's apparent advantage by the amount of experience we have with it, and so address the well-known tension between exploiting an estimate and trusting it. We also read, but do not maintain, the active knowledge graph F_t at each step. The knowledge graph is updated according to the exogenous arrival process P_F described in Section 7.3.3, and we have no control over which facts arrive when; we observe the current active set at each step and use it to compile the current aspiration.

7.5.2 Computing the Risk-Sensitive Value

Given the current state s_t and the current active fact set F_t, two quantities are computed at each step before action selection. The first is the aspiration itself,

$$\rho_t \;=\; C(s_t, F_t)$$

a single real number obtained from the compiler in O($|F_t|$) time as a sum of per-fact contributions. The second is the per-action **risk-sensitive value**

$$\mathrm{RS}_t(s_t, a) \;=\; \tau(s_t, a) \cdot [\, Q(s_t, a) - \rho_t \,]$$

defined as the product of the reliability count and the advantage Q(s_t, a) − ρ_t, where we use the term advantage to denote the gap between the current Q estimate and the current aspiration. The sign of RS_t matches the sign of the advantage because τ is non-negative, and the magnitude grows linearly with the reliability count, so an action with both a substantial positive advantage and many prior visits attains a large RS value, while an action with the same apparent advantage but few prior visits attains only a small RS value. An action whose Q estimate exceeds (or equals) the aspiration is called **satisficing** at s; otherwise it is called sub-satisficing. At the very start of training τ(s, a) = 0 for every a, so RS_t(s_t, a) = 0 uniformly across actions and ties are resolved at random, producing effectively uniform exploration until visit counts accumulate. This early behavior is desirable: we do not commit to an action on the basis of a Q estimate we have not yet had a chance to validate.

7.5.3 Selecting an Action

In CA-RS-Q we apply an epsilon-greedy rule to the RS values rather than to the raw Q values. With probability ε we select an action uniformly at random from A. With probability $1 - \varepsilon$ we select

$$a_t \;=\; \arg\max_{a \in \mathcal{A}} \mathrm{RS}_t(s_t, a) \tag{7.23}$$

with ties broken uniformly at random. The exploration parameter ε is a small positive constant, required to be strictly positive so that every state-action pair is visited infinitely often in expectation, a condition we will need for the convergence guarantee in Section 7.6. The rule has two properties of note. First, it does not exclude any action from consideration. The aspiration is not a constraint that filters the action space; an action whose Q estimate falls below the aspiration is still a candidate, and it will be selected when its reliability-weighted advantage is the largest among all available actions. This avoids the failure mode of strict constraint-based methods, in which a tightening of the aspiration can leave no satisficing action available and force undefined behavior. Second, the action-selection rule depends on Q, τ, and ρ jointly, and the joint dependence is multiplicative through τ and additive through ρ. The resulting preference ordering over actions changes when any of these quantities changes, but in a way that is interpretable: a change in ρ shifts the zero of the advantage uniformly across actions, a change in τ rescales the contribution of one action without affecting the others, and a change in Q at one (s, a) pair affects only that action's RS value.

7.5.4 Updating the Value Estimate and the Reliability Count

After we select action a_t, execute it in the environment, and observe the reward r_t and the next state s_{t+1}, two updates follow. The action-value estimate is updated by the standard one-step Q-learning rule,

$$Q(s_t, a_t) \;\leftarrow\; Q(s_t, a_t) \;+\; \alpha_t \cdot \Big[\, r_t + \gamma \cdot \max_{a' \in \mathcal{A}} Q(s_{t+1}, a') \;-\; Q(s_t, a_t) \,\Big], \tag{7.24}$$

where α_t is the learning rate at step t, required to satisfy the Robbins-Monro conditions $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$ for the convergence guarantee of Section 7.6.1. The reliability count is

incremented:

$$\tau(s_t, a_t) \leftarrow \tau(s_t, a_t) + 1 \tag{7.25}$$

No other entries of Q or τ are modified at this step. The aspiration ρ_t is not stored from one step to the next; we recompute it from F_t at the start of every step. This recomputation is the mechanism by which CA-RS-Q responds to changes in the knowledge graph between steps.

7.5.5 The Role of the Aspiration

The aspiration ρ_t appears in the action-selection rule of Section 7.5.3 but not in the Q-update of Section 7.5.4. This separation is the central design decision of CA-RS-Q. Were the aspiration embedded in the Q-update, the value estimate would learn the aspiration-adjusted return rather than the true return, and any subsequent change in the aspiration would render the existing Q-table inconsistent with the new target. At each fact arrival we would then have to relearn from scratch, which defeats the purpose of an evolving knowledge graph. Keeping the Q-update aspiration-free avoids this failure mode at the cost of giving up explicit optimization of the aspiration-adjusted return, a cost we acknowledge and discuss in Section 7.8. The benefit is threefold and substantive. First, the Q estimates converge to Q^* of the underlying MDP under standard conditions, and Theorem 1 makes this precise by reducing the convergence question to the classical Watkins-Dayan [4] result. Second, the action we select responds to a changed knowledge graph at the very next time step, without requiring any update of the value estimates and without any retraining cost. Third, the contribution of each active fact to a given decision equals exactly that fact's contribution to the compiled aspiration at that decision, providing the structural basis for the audit trail we develop in Section 7.7.7.

A practical consequence of this design is that CA-RS-Q is not a constrained optimizer in the sense of constrained MDP methods, nor is it a reward-shaping method in the sense of potential-based shaping [24]. The aspiration does not modify the reward function, and it does not impose a hard constraint on the policy. It modifies the action-selection rule at each step, with the effect that the policy preferred by CA-RS-Q is biased toward actions whose Q estimates exceed the aspiration, and the strength of this bias scales with the reliability count of the candidate actions. The Q estimates themselves remain unbiased estimators of Q^* in the limit guaranteed by Theorem 1. In Sections 6 and 7, we develop both the theoretical and empirical consequences of this construction.

7.5.6 Behavior Under an Evolving Knowledge Graph

As the knowledge graph evolves, the per-step behavior of CA-RS-Q follows directly from the construction above. At each step we read the current active fact set, compute the aspiration as a sum of per-fact contributions, evaluate the RS value for each action under the current Q and τ tables, and select an action by the epsilon-greedy rule. The arrival of a new fact at the start of step t modifies the active set F_t, which modifies ρ_t through the per-fact sum, which modifies RS_t through the advantage, which then modifies the argmax. We carry this modification through the per-step computation without any retraining and without any reset of internal state. The expiration of an existing fact at the end of its validity window has the symmetric effect: the corresponding contribution is no longer part of the per-fact sum, the aspiration is higher than it was during the fact's validity, and the RS values for sub-satisficing actions become less negative. When the knowledge graph remains constant for a sufficiently long stretch (we make this precise in Section 7.6.1), the Q estimates continue to converge toward Q^* and the satisficing rate stabilizes at the level predicted by Theorem 2. When the knowledge graph changes after this convergence has been substantially achieved, the aspiration changes and the selected action changes, and Theorem 3 bounds the magnitude of the resulting change in policy value.

7.5.7 Per-Step Cost

We close this section with the cost of a single step. Per step, we perform one compiler call (O(|F|) in the flat setting, and O(|$F_s(G_t)$|) in the graph-structured setting); |A| risk-sensitive value computations, each O(1); one argmax over |A| values, O(|A|); one Q-update, O(|A|), including the max over the next state's Q row; and one reliability-count increment, O(1). The total per-step cost is O(|A| + |F|), which matches standard Q-learning plus a compiler call linear in the active fact set, and which is negligible relative to the cost of environment interaction in the experiments of Section 7.7.

7.5.8 Relationship to Prior Algorithms

CA-RS-Q reduces to the risk-sensitive satisficing Q-learning algorithm of Takahashi et al. [9] in the special case in which the active fact set is empty at every time step, so that $\rho_t(s) = \rho_0(s)$ is a fixed designer-specified baseline rather than a time-varying compiled quantity. CA-RS-Q further reduces to standard Q-learning in the special case in which the reliability count is replaced by a

constant across actions and the aspiration is the same across actions: the argmax over τ(s, a) · (Q(s, a) − ρ) then coincides with the argmax over Q(s, a), so the action-selection rule of standard Q-learning is recovered. Within this hierarchy of algorithms, CA-RS-Q occupies a specific position. The value-learning component is standard. The reliability-weighted satisficing logic in action selection is from Takahashi. The structured, graph-grounded compilation of the aspiration is our contribution. The theorems of Section 7.6 quantify what we obtain from this construction in terms of convergence, satisficing rate, and robustness under perturbations of the knowledge graph.

7.6. Theoretical Results

We establish four theorems. In Theorem 1 we establish convergence of the Q estimates under an eventually stable knowledge graph. In Theorem 2 we establish a finite-time, high-probability bound on the per-episode satisficing rate under a separation gap assumption. In Theorem 3 we establish a piecewise bound on the change in policy value under perturbations of the knowledge graph treated as a flat fact set. In Theorem 3-G we extend Theorem 3 to the graph-structured setting of Section 7.4.3. We prove Theorems 1, 3, and 3-G in full and sketch Theorem 2; the full proof of Theorem 2 is structurally a restatement of the Even-Dar and Mansour bound under the satisficing decision rule and we defer it to the supplementary material. The proofs of Theorems 1 and 2 reduce to the classical Q-learning convergence result of Watkins and Dayan [4] and the sample-complexity result of Even-Dar and Mansour [36]; the reductions are direct consequences of the design choice in Section 7.5.5 to keep the value-learning component independent of the aspiration. Theorems 3 and 3-G require new analysis and admit a piecewise structure, which we develop explicitly.

7.6.1 Theorem 1: Convergence Under an Eventually Stable Knowledge Graph

We require four assumptions for the convergence result. The first three are standard in the analysis of Q-learning. We assume bounded rewards, denoted **(BR)**: there exists $R_{\max} < \infty$ such that $|R(s,a)| \leq R_{\max}$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. We assume infinite visitation, denoted **(V)**: every state-action pair (s,a) is visited infinitely often almost surely, a condition ensured by the use of an epsilon-greedy exploration rule with $\varepsilon > 0$ in Section 7.5.3. We assume Robbins-Monro learning rates, denoted **(RM)**: $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$.

Read precisely, for (RM) we require the conditions to hold along the subsequence of steps at which each state-action pair is visited; the implementation uses $\alpha_t = 1/n_{s_t,a_t}^{0.7}$ with $n_{s,a}$ the per-pair visit count, which satisfies the per-pair form. The global-t phrasing above is standard shorthand.

The remaining assumption is specific to the CA-MDP setting. We denote it **(ES)** for **eventual stability with dynamics-neutral fact set**: there exists a finite $T^* < \infty$ and a fact set F^*, such that $F_t = F^*$ for all $t \geq T^*$ almost surely, **and** the env's effective transition kernel and reward function under F^* coincide with those of the underlying MDP $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. Assumption (ES) permits the knowledge graph to change arbitrarily during the prefix $t < T^*$ and requires only that the changes terminate at some finite (but unconstrained) time, with the post-stabilization fact set being dynamics-neutral. In the inventory benchmark, dynamics-neutrality holds when $F^* = \emptyset$. A non-empty F^* is dynamics-neutral only if no active fact type modulates the env's transition kernel or reward and no accompanying change to the env's cost parameters is in effect; the regulator-shock regimes of Experiments 6 and 8 are therefore not dynamics-neutral, because each raises the env's backlog-cost coefficient (cost_backlog $\times$ 5) at the same time the regulator fact is injected. In our Experiment 1 (Section 7.7.2) we engineer $F^* = \emptyset$ by halting the news stream at episode 200. Assumption (ES) does not constrain T^*, and the value of T^* need not be known to the algorithm. In Experiments 2 through 9 we train under non-stationary or persistent news with active dynamics modulation. Those regimes lie outside the strict scope of Theorem 1, and we report them as empirical robustness results rather than as convergence verifications.

Theorem 1 (Per-regime convergence). *Under assumptions (BR), (V), (RM), and (ES), the action-value estimates produced by CA-RS-Q satisfy*

$$Q_t(s,a) \;\to\; Q^*(s,a) \quad \text{almost surely as } t \to \infty \tag{7.26}$$

for every $(s,a) \in \mathcal{S} \times \mathcal{A}$, where Q^* is the optimal action-value function of the underlying MDP $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ defined in Section 7.3.1, evaluated in the absence of facts.

The limit Q^* is the optimal value function of the underlying fact-free MDP, fixed throughout the argument. The Q-update is independent of the aspiration ρ_t, so the value estimates track Q^* regardless of the action-selection rule that biased exploration during the prefix $t < T^*$. The aspiration $\rho^*(s) = C(s, F^*)$ influences action selection through the RS rule but does not appear in the Q-update; this property is the formal counterpart of the design choice recorded in Section 7.5.5.

Remark (benchmark realism). Our inventory benchmark env additionally lets active facts perturb the transition kernel and reward function during the prefix $t < T^*$ (capturing real supply-chain effects such as supplier-disruption-induced shortages and demand-surge-induced cost spikes). This perturbation is a property of the empirical environment, not of the formal CA-MDP model: the underlying MDP and its Q^* are fact-free. Assumption (ES) is to be interpreted accordingly, requiring the post-stabilization regime to coincide with the underlying MDP. In our Experiment 1 (Section7.2), this is engineered by halting the news stream at episode 200, so that $F^* = \emptyset$ and the env-level perturbation vanishes from time $T^* = 200$ onward.

Proof. For $t \geq T^*$, by (ES) the active fact set is $F_t = F^*$. The compiled aspiration $\rho_t = C(s_t, F^*)$ is then a fixed function of state, since C is a deterministic function of (s, F) by its signature. The Q-update in step (5) of Algorithm 7.1 depends only on the observed tuple (s_t, a_t, r_t, s_{t+1}) and the current Q-table; it does not depend on ρ_t. For all $t \geq T^*$, the Q-table evolves according to the standard one-step Q-learning rule in the underlying MDP $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, with action selection given by the epsilon-greedy RS rule. By (V), every state-action pair continues to be visited infinitely often; by (RM), the learning rates satisfy the Robbins-Monro conditions; by (BR), rewards are bounded by $R_{\max}$. The Watkins-Dayan convergence theorem [4, 32, Section5.6] applies to the post-T^* tail: the iterates of Q-learning under (V), (RM), and (BR) converge to Q^* almost surely, irrespective of the action-selection rule. The pre-T^* trajectory contributes only finitely many updates to the Q-table, which Watkins-Dayan accommodates as an arbitrary initialization for the post-T^* tail. Hence $Q_t(s, a) \to Q^*(s, a)$ almost surely for every $(s, a) \in \mathcal{S} \times \mathcal{A}$. ■

For the proof, we rely on Watkins-Dayan as an established result; rigorous derivations are in the original paper and in Bertsekas and Tsitsiklis [32, Section5.6]. The underlying MDP $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ is stationary throughout the argument, and the Q-update is independent of the aspiration ρ_t. Theorem 1 establishes convergence of the value estimates to Q^* in the post-T^* tail under (ES); behavior during the non-stationary prefix $t < T^*$ contributes only finitely many updates and does not affect the limit. None of the four compiler axioms (A1) through (A4) is required for the convergence argument; they are required for Theorems 2, 3, and 3-G below.

7.6.2 Theorem 2: Finite-Time Satisficing Rate

In Theorem 1 we make an asymptotic claim about the limit of the Q estimates, and we say nothing there about the rate at which the agent selects satisficing actions during training. In Theorem 2 we establish a finite-time, high-probability bound on that rate. We say that an action a is **satisficing in the normative sense** at state s under aspiration ρ^* if $Q^*(s,a) \geq \rho^*(s)$, where Q^* is the optimal action-value function of the underlying MDP. This is distinct from the **operational satisficing** notion used in Section 7.5.2, where an action a is satisficing at state s under the agent's current Q-table if $Q(s,a) \geq \rho_t(s)$; the operational notion uses the agent's running estimate, the normative notion uses the optimal reference. Theorem 2 below establishes a bound on the rate at which actions selected by the agent are satisficing in the normative sense.

Remark (the satisficing reference under benchmark env perturbation). The satisficing condition is stated against Q^* of the underlying fact-free MDP, and we compute Q^* once by value iteration with the active fact set held at $\emptyset$. In our benchmark env (Section7.1), during stationarity assumption (S) with a non-empty fact set F^*, active facts perturb the transition kernel and reward function as a benchmark-realism layer (see the corresponding remark in Section6.1). Under such perturbation, the agent's Q-table during training tracks the optimum of the F^*-modulated env rather than Q^* of the underlying MDP. We retain Q^* of the underlying MDP as the static satisficing reference because the compiler's per-fact contribution functions ψ_θ are designed to approximately offset the env-level value shift induced by each fact type — adverse facts lower ρ; favorable facts raise ρ — so that the satisficing condition $Q^*(s,a) \geq \rho_t$ against the fact-adjusted aspiration remains a meaningful criterion in the perturbed env. The empirical satisficing rate reported in Section 7.7.3 is measured against this static reference, and the gap diagnostic in Table 7.4 captures the residual margin separating this approximation from a fully fact-indexed analysis.

We rely on three assumptions in addition to those we adopt for Theorem 1 when we derive this result. We replace (ES) with a stronger condition, denoted **(S)** for stationarity: $F_t = F^*$ for all $t \geq 0$. Stationarity is strict from the first step, so the Even-Dar and Mansour [36] sample-complexity result applies without modification. We let $\rho^*(s) = C(s, F^*)$ denote the stationary aspiration.

We add a separation gap condition, denoted **(G+)**, which has two parts. The first part, denoted **(G+.i)**, requires the existence of $g > 0$ such that at every state s, at least one action $a^*(s) \in \mathcal{A}$

satisfies $Q^*\big(s, a^*(s)\big) - \rho^*(s) \geq g$. A satisficing action exists at every state, with positive margin at least g. The second part, denoted **(G+.ii)**, requires that every action $a \in \mathcal{A}$ satisfies either $Q^*(s,a) - \rho^*(s) \geq g$ (clearly satisficing) or $Q^*(s,a) - \rho^*(s) \leq -g$ (clearly non-satisficing); no actions occupy the open band $(-g, g)$. Together, (G+.i) and (G+.ii) ensure that satisficing is achievable everywhere and that the satisficing classification of every action is preserved under finite-sample Q-estimation error of magnitude up to $g/2$.

Theorem 2 (Finite-time satisficing rate). *Under assumptions (BR), (V), (V+), (RM), (S), and (G+) with margin g, for any $\delta \in (0,1)$, there exists a finite $N_0 = N_0(\delta, |\mathcal{S}|, |\mathcal{A}|, \gamma, g, \varepsilon, R_{\max}, L_{\text{cov}})$ such that for all $N \geq N_0$,*

$$\mathbb{P}[\text{ on episode } N, \text{ every non-exploratory action selected by CA-RS-Q is satisficing at its state }] \geq 1 - \delta.$$

Here **(V+)** *denotes the bounded-covering-time assumption: there exists $L_{\text{cov}} < \infty$, a property of the underlying MDP and the exploration rule alone, such that under the epsilon-greedy exploration rule with rate $\varepsilon > 0$, the expected first-passage time to every state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ from any starting state is at most L_{cov}. The bound N_0 is polynomial in $|\mathcal{S}|$, $|\mathcal{A}|$, L_{cov}, $R_{\max}/(1-\gamma)$, $1/g$, $1/\varepsilon$, and logarithmic in $1/\delta$.*

Proof sketch. By Even-Dar and Mansour [36], under (V), (V+), (RM), and (S), Q-learning admits a polynomial-sample PAC bound: for any $\eta > 0$ and $\delta_1 \in (0,1)$, there exists $N_1 = N_1(\eta, \delta_1, |\mathcal{S}|, |\mathcal{A}|, L_{\text{cov}}, \gamma, \varepsilon, R_{\max})$, polynomial in its arguments, such that $\mathbb{P}[\,\| Q_N - Q^* \|_\infty \leq \eta\,] \geq 1 - \delta_1$ for all $N \geq N_1$.

Set $\eta = g/2$. By (G+.i), for any satisficing action $a^*(s)$, $Q^*\big(s, a^*(s)\big) - \rho^*(s) \geq g$. On the event $\{\| Q_N - Q^* \|_\infty \leq g/2\}$, the inequality $Q_N\big(s, a^*(s)\big) - \rho^*(s) \geq g/2 > 0$ holds for every satisficing action. By (G+.ii), every non-satisficing action a' satisfies $Q^*(s,a') - \rho^*(s) \leq -g$, so on the same event $Q_N(s,a') - \rho^*(s) \leq -g/2 < 0$. Sample-complexity error of magnitude up to $g/2$ therefore preserves every action's satisficing classification in the Q estimates.

Under (V) and (V+), the reliability count $\tau(s,a)$ is incremented each time the pair (s,a) is taken. Let T_{cov} denote the random first-passage time required for every pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ to be visited at least once under the epsilon-greedy exploration rule. By (V+), $\mathbb{E}[T_{\text{cov}}] \leq L_{\text{cov}}$. By Markov's inequality, $\mathbb{P}[T_{\text{cov}} \geq K \cdot L_{\text{cov}}] \leq 1/K$ for any $K > 0$. Choosing $K = 2/\delta_2$ gives $\mathbb{P}[T_{\text{cov}} \leq (2/\delta_2) \cdot$

$L_{\text{cov}}] \geq 1 - \delta_2$. Hence the event $\{\tau_N(s,a) \geq 1 \text{ for every } (s,a) \in \mathcal{S} \times \mathcal{A}\}$ occurs in $N_2 = \mathcal{O}\big((L_{\text{cov}}/\delta_2) \cdot (1/\varepsilon)\big)$ episodes (recalling that each episode contributes a bounded number of steps and ε-exploration occurs at rate ε) with probability at least $1 - \delta_2$. On this event, $\tau(s,a) \geq 1$ for every (s,a).

On the intersection of the two events $\{\| Q_N - Q^* \|_\infty \leq g/2\}$ and $\{\tau_N(s,a) \geq 1 \text{ for every } (s,a)\}$, the RS values satisfy $\text{RS}_N\big(s, a^*(s)\big) \geq 1 \cdot (g/2) > 0$ for every satisficing action and $\text{RS}_N(s,a') \leq \tau(s,a') \cdot (-g/2) < 0$ for every non-satisficing action. The argmax over RS_N at state s therefore selects an action with positive RS value, which by the inequalities above is satisficing.
Setting $\delta_1 = \delta_2 = \delta/2$, the union of the two events $\{\| Q_N - Q^* \|_\infty \leq g/2\}$ and $\{\tau_N(s,a) \geq 1 \text{ for every } (s,a)\}$ has probability at least $1 - \delta$. The required N_0 is the larger of $N_1(g/2, \delta/2, |\mathcal{S}|, |\mathcal{A}|, L_{\text{cov}}, \gamma, \varepsilon, R_{\max})$ and $N_2(\delta/2)$, each polynomial in its arguments. ■
The polynomial form is

$$N_0 \;=\; \mathcal{O}\left(\frac{L_{\text{cov}} \cdot |\mathcal{S}| \cdot |\mathcal{A}| \cdot (R_{\max}/(1-\gamma))^2 \cdot \log(1/\delta)}{g^2 \cdot \varepsilon}\right) \quad (7.27)$$

The bound applies to the argmax branch of action selection. The epsilon-exploratory branch selects actions uniformly at random with probability ε at each step, and these selections may be non-satisficing. We report the empirical satisficing rate under both branches in Section 7.7.3. Assumption (G+.ii) is an idealization. In our concrete benchmarks, a small fraction of (s, a) pairs lie in the open band (−g, g) and contribute to a residual satisficing failure rate not captured by the bound.

7.6.3 Theorem 3: Robustness Under Flat Knowledge Graph Perturbation

In Theorems 1 and 2 we characterize training under a stationary or eventually stationary knowledge graph. In Theorem 3 we characterize how a change in the knowledge graph affects the resulting policy. Theorem 3 characterizes the effect of changes in the knowledge graph on the resulting policy. We measure the effect by the difference in policy value in the underlying MDP, as a function of the perturbation distance d_F(F, F') defined in Section 7.3.3.

We fix the underlying MDP, a compiler C satisfying (A2) with Lipschitz constant L_C, and a reliability table τ. The reliability table is treated as fixed for the purpose of the analysis and can be interpreted as the table at any specific point in training; the result does not depend on which point.

For each fact set F, the **RS-greedy policy under F**, evaluated with the true Q^*, is

$$\pi_F(s) \;=\; \arg\max_{a\in\mathcal{A}} \tau(s,a)\cdot[\,Q^*(s,a) - C(s,F)\,] \tag{7.28}$$

We denote the value of the idealized Q^*-RS-greedy policy under fact set F by V^{π_F}, evaluated in the underlying MDP. In Theorem 3 we bound the change in this idealized value under perturbations of F. Robustness of the deployed CA-RS-Q policy — which acts on the learned estimate Q_N rather than Q^* — additionally inherits the finite-sample error of Q_N via the analysis in Theorem 2. We compare V^{π_F} and $V^{\pi_{F'}}$ for two arbitrary fact sets F and F'.

We require a margin condition for this result, denoted (M): for the specific fact sets F and F' under analysis in the theorem, there exists $m_{F,F'} > 0$ such that at every state $s \in \mathcal{S}$, the gap between the highest and second-highest RS values at s, computed with τ and aspiration $C(\cdot,F)$ using the true Q^*, is at least $m_{F,F'}$, and likewise for $C(\cdot,F')$. Assumption (M) is the analog of (G+) at the level of RS values rather than raw advantages. It is satisfied in non-degenerate MDPs and we verify it empirically in Section 7.7.4. The "policy-switch threshold" introduced after the theorem statement is implicitly indexed by the same pair (F,F').

Theorem 3 (Robustness under flat KG perturbation). *Under (A2) with Lipschitz constant L_C, (BR) with bound $R_{\max}$, and assumption (M), for any state s and any two fact sets F, F',*

$$|V^{\pi_F}(s) - V^{\pi_{F'}}(s)| \;\le\; \tfrac{2R_{\max}}{1-\gamma}\cdot \mathbf{1}\{\,\tau_{\max}\cdot L_C \cdot d_F(F,F') \;\ge\; m\,\} \tag{7.29}$$

where $\tau_{max} = max_{s,a}\tau(s,a)$ and $\mathbf{1}\{\cdot\}$ denotes the indicator function.

The bound is piecewise. When $\tau_{\max}\cdot L_C \cdot d_F(F,F') < m$, the two policies coincide at every state and the value difference vanishes. When $\tau_{\max}\cdot L_C \cdot d_F(F,F') \ge m$, the trivial worst-case bound $2R_{\max}/(1-\gamma)$ applies. A smooth Lipschitz bound on V in terms of d_F does not hold in general because policy switches are discrete events at which the value can change by a non-vanishing amount.

Proof. By (A2), for any state s, |C(s, F) − C(s, F')| ≤ L_C · d_F(F, F'). For any action a at state s,

$$|\mathrm{RS}_F(s,a) - \mathrm{RS}_{F'}(s,a)| \;=\; \tau(s,a)\cdot|C(s,F) - C(s,F')| \;\le\; \tau(s,a)\cdot L_C \cdot d_F(F,F') \tag{7.30}$$

Let $\Delta = L_C \cdot d_F$(F, F'). By (M), the gap between the highest and second-highest RS values at every state, under either aspiration, is at least m.

In Case A, suppose $\Delta \cdot \tau_{\max} < m$. Write $\delta_C(s) = C(s, F') - C(s, F)$ for the aspiration shift at state s, so $|\delta_C(s)| \leq L_C \cdot d_F(F, F') = \Delta$ by (A2). The shift $\delta_C(s)$ is common to all actions at state s and enters $\mathrm{RS}_F(s, a)$ scaled only by $\tau(s, a)$. The *differential* shift between any two actions a, b at state s is therefore $|\tau(s, a) - \tau(s, b)| \cdot |\delta_C(s)| \leq \tau_{\max} \cdot \Delta < m$, which is strictly smaller than the gap m guaranteed by (M). The ranking of RS values is therefore preserved at every state, so $\pi_F(s) = \pi_{F'}(s)$ for all s. The two policies coincide. Evaluated in the underlying MDP, $V^{\pi_F}(s) = V^{\pi_{F'}}(s)$ for all s, and the bound holds with value zero on the right.

In Case B, suppose $\Delta \cdot \tau_{\max} \geq m$. The ranking may switch at some states and the resulting policies may differ. Any two policies π, π' in an MDP with rewards bounded by $R_{\max}$ satisfy

$$|V^{\pi}(s) - V^{\pi'}(s)| \;\leq\; \frac{2R_{\max}}{1-\gamma} \tag{7.31}$$

because each value function is bounded in $[-R_{\max}/(1-\gamma),\, R_{\max}/(1-\gamma)]$. The bound holds with the right-hand side equal to $2R_{\max}/(1-\gamma)$.

The two cases combine into the piecewise bound stated in the theorem. ■

The quantity $m/(\tau_{\max} \cdot L_C)$ is the **policy-switch threshold**, the value of $d_F(F, F')$ at which the inequality $\Delta \cdot \tau_{\max} \geq m$ becomes attainable. Perturbation distances strictly below this threshold leave the policy unchanged at every state. Perturbations at or above the threshold may produce policy switches that we cannot bound below the trivial worst case. As $\tau_{\max}$ grows, the threshold decreases, so a reliability table accumulated over more training is sensitive to smaller perturbation distances. We measure the empirical value gap in Section 7.7.4 and find that the actual gap is a small fraction of the trivial bound on both benchmarks, even though our perturbations were chosen to exceed the threshold.

7.6.4 Theorem 3-G: Robustness Under Graph-Structured Perturbation

In the graph-structured extension of Section 7.4.3 we replace the flat fact set with a triple G=(V,E,F), restrict the compiler to operate on the state-relevant fact set F_s(G) we define by the 1-hop traversal from the state's entities, and replace the fact-set distance dF with the graph distance

$$d_G(G, G') \;=\; w_V \cdot |V \,\triangle\, V'| \;+\; w_E \cdot |E \,\triangle\, E'| \;+\; d_F(F, F') \tag{7.32}$$

The Lipschitz axiom for the graph-structured setting, denoted **(A2-G)**, states that there exists $L_G > 0$ such that for every state s and every pair of graphs G, G',

$$|C(s,G) - C(s,G')| \;\leq\; L_G \cdot d_G(G,G') \tag{7.33}$$

We establish (A2-G) as a consequence of (A2) plus the boundedness property (P1) of the concrete compiler in Section 7.4.4, with

$$L_G \;=\; \max\{\, L_C,\; c_{\max} \cdot \psi_{\max} \cdot \varphi_{\max} \,\} \tag{7.34}$$

where L_C is the flat-KG Lipschitz constant of (A2), $c_{\max} \leq 1$ bounds the per-fact confidence, $\psi_{\max}$ = max_{f, s} |ψ_{type(f)}(f, s)| bounds the per-fact contribution magnitude, and $\varphi_{\max}$ is the maximum number of facts attached to any single entity in F ∪ F'.

The two terms in the maximum cover the two sources of perturbation in d_G. The first term, L_C, is the Lipschitz constant inherited from (A2) and controls the contribution of d_F(F, F') to d_G(G, G'). The second term, $c_{\max} \cdot \psi_{\max} \cdot \varphi_{\max}$, controls the contribution of node and edge differences: a single new edge can pull at most $\varphi_{\max}$ facts into the 1-hop neighborhood of some state, and each such fact contributes at most $c_{\max} \cdot \psi_{\max}$ to the aspiration under the concrete compiler. Taking the maximum covers all three perturbation types. A detailed derivation is in the supplementary material; the central observation is that node and edge changes affect the aspiration only through their effect on F_s(G), and the per-fact contribution is bounded.

The margin condition for the graph-structured setting, denoted **(M-G)**, is identical to (M) but applied to RS values computed using F_s(G) under each graph rather than the full fact set.

Throughout this subsection, V denotes the vertex set of the knowledge graph (as introduced in Section3.3), while $V^{\pi}(s)$ denotes the policy value function from Section3.1; the two symbols are distinct and we rely on context to indicate which is meant.

Theorem 3-G (Robustness under graph-structured KG perturbation). *Under (A2-G) with Lipschitz constant $L_G > 0$, (BR) with bound $R_{\max}$, and assumption (M-G), for any state s and any two graphs G, G',*

$$|V^{\pi_G}(s) - V^{\pi_{G'}}(s)| \;\leq\; \frac{2R_{\max}}{1-\gamma} \cdot \mathbf{1}\{\, \tau_{\max} \cdot L_G \cdot d_G(G,G') \;\geq\; m \,\} \tag{7.35}$$

Proof. Identical in structure to the proof of Theorem 3, with d_F replaced by d_G, L_C replaced by L_G, and (M) replaced by (M-G) throughout. The Case A differential-shift argument from the proof of Theorem 3 and the Case B trivial bound carry through unchanged.■

Relationship to Theorem 3. Theorem 3-G and Theorem 3 are not strictly nested. Theorem 3-G specializes to a special case of Theorem 3 in the degenerate-graph limit $V = V' = \emptyset$, in which case the 1-hop neighborhood is empty at every state and the state-relevant fact set $F_s(G) = \{f \in F : \mathrm{ent}(f) = \emptyset\}$ retains only the graph-wide facts. In this limit the second term in the maximum defining L_G becomes vacuous (no entity-referencing facts can be perturbed in or out via node/edge changes), and the resulting bound matches Theorem 3 restricted to graph-wide facts. Theorem 3 itself, however, holds for any fact set $F \subseteq \mathcal{F}$, including entity-referencing facts; the graph-structured theorem extends robustness analysis to perturbations of the graph carrier of those facts, which Theorem 3 does not address. The two theorems thus cover overlapping but non-identical territory, with the shared special case described above.

Lemma 6.4 (Admissibility of the concrete compiler). *The concrete compiler of Section 7.4.4,* $C(s,F) = \rho_0(s) + \sum_{f\in F} c\,(f) \cdot \psi_{\mathrm{type}(f)}(f,s)$*, satisfies axioms (A1), (A2), (A3), and (A4) under properties (P1)–(P3) of* $\{\psi_\theta\}_{\theta\in\Theta}$*, with Lipschitz constant*

$$L_C \;\le\; M_{\max}/v_{\min}, \tag{7.36}$$

where $M_{\max} = \max_\theta M_\theta$ *and* $v_{\min}$ *is the lower bound on severity magnitude from Section 7.3.3.*

Proof. (A1) is immediate: $C(s,\emptyset) = \rho_0(s) + 0 = \rho_0(s)$. For (A2), for any $s \in \mathcal{S}$ and any $F, F' \subseteq \mathcal{F}$,

$$|C(s,F) - C(s,F')| \;\le\; \sum_{f\in F\triangle F'} c\,(f) \cdot \left|\psi_{\mathrm{type}(f)}(f,s)\right| \;\le\; \sum_{f\in F\triangle F'} c\,(f) \cdot M_{\max}$$
$$\le\; \frac{M_{\max}}{v_{\min}} \cdot d_F(F,F'), \tag{7.37}$$

where the first inequality uses (P1) and the second uses $|v(f)| \ge v_{\min}$. For (A3), if $F \subseteq F'$ and every $f \in F'\backslash F$ has $\tau_f = +1$, then by (P2) each $\psi_{\mathrm{type}(f)}(f,s) \le 0$, so $C(s,F') = C(s,F) + \sum_{f\in F'\backslash F} c\,(f) \cdot \psi_{\mathrm{type}(f)}(f,s) \le C(s,F)$. The loosening case is symmetric. For (A4), define

$\Phi(F,s) = \sum_{f \in F} c(f) \cdot \psi_{\text{type}(f)}(f,s)$; then $C(s,F) = \rho_0(s) + \Phi(F,s)$, $\Phi(\emptyset,s) = 0$, and Φ is Lipschitz in F with the constant derived for (A2). ■

Summary: Theorem 1 establishes convergence of the Q estimates to the optimal action-value function of the underlying MDP under eventual stability of the knowledge graph. Theorem 2 establishes a polynomial-sample, high-probability bound on the per-episode satisficing rate of the non-exploratory action selections under stationarity and the separation gap condition (G+). Theorem 3 establishes a piecewise bound on the change in policy value under perturbations of the knowledge graph treated as a flat fact set. Theorem 3-G generalizes Theorem 3 to perturbations of the graph-structured knowledge graph under the metric d_G; in the degenerate-graph limit $V = V' = \emptyset$, it specializes to a graph-wide-facts-only restriction of Theorem 3 rather than recovering Theorem 3 in full, as detailed in the "Relationship to Theorem 3" remark of Section 7.6.4. The four results collectively characterize CA-RS-Q's behavior in three regimes: limiting behavior under stable structured information, finite-time behavior under stationary structured information, and stability under bounded changes to the structured information. We verify all four results empirically in Section 7.7.

Table 7.2: Summary of Theorems

Result	Assumptions used	Conclusion
Theorem 1	((BR), (V), (RM), (ES)	$Q_t \to Q^*$ almost surely
Theorem 2	(BR), (V), (V+), (RM), (S), (G+)	$\mathbb{P}$[episode N satisficing] $\geq 1 - \delta$ for $N \geq N_0$ polynomial
Theorem 3	(A2), (BR), (M), fixed τ	Piecewise bound on $\| V^{\pi_F} - V^{\pi_{F'}} \|$ with threshold $m/(\tau_{\max} \cdot L_C)$
Theorem 3-G	(A2-G), (BR), (M-G), fixed τ	Piecewise bound on $\| V^{\pi_G} - V^{\pi_{G'}} \|$ with threshold $m/(\tau_{\max} \cdot L_G)$

7.7 Results and Empirical Evaluation

We verify the four theoretical results of Section 7.6 on a family of inventory-management benchmarks, and we then carry our design objective for the framework, protection against catastrophic outcomes under structural shock, to a problem scale large enough that the deployment-relevant consequences become visible. We use three benchmarks. The first, **Benchmark A**, is the single-product benchmark introduced as the running example in Section 7.3.2. In the second, **Benchmark B**, **we extend** the same problem to two products. In the third, Benchmark C, **we extend** it to three products and is approximately 264 times the state-action size of Benchmark B. All three share the fact schema of Section 7.3.3, the compiler form of Section 7.4.4, and the algorithm of Section 7.5; they differ in the size of the state and action spaces, in the structure of the underlying knowledge graph, and in the per-type compiler coefficients and baseline.

We use the three benchmarks in two distinct roles. Benchmarks A and B are the theorem-verification benchmarks. They are small enough that every numerical effect we report can be inspected directly by a reader who wishes to reproduce the underlying computation, and we use them to verify Theorems 1 through 3 in parallel. Benchmark C is the scaling benchmark. Convergence to Q^* (Theorem 1) and the satisficing-rate diagnostic (Theorem 2) require the true optimal value function as a reference, a settled quantity that the theory already establishes, so re-running those two diagnostics on a 287,496-state Markov decision process would add computational cost without adding insight, and we do not repeat them on Benchmark C. Instead, we carry the robustness result (Theorem 3) and the catastrophe-protection finding to Benchmark C, because those are the results whose practical consequence we expect to grow with problem size.

In Section 7.7.1 we specify all three benchmarks. In Sections 7.7.2 through 7.7.4 we verify Theorems 1, 2, and 3, presenting Benchmark A and Benchmark B in parallel and extending the robustness verification to Benchmark C. In Section 7.7.5 we report the catastrophe-protection finding across all three benchmarks, isolate the mechanism through a compiler ablation, and report a paired significance test. In Section 7.7.6 we collect the cross-benchmark comparison in a single table. In Section 7.7.7 we walk through a worked decision card to demonstrate the auditability property we claim in Section 7.5.5. In Section 7.7.8 we summarize the findings. Unless we state otherwise, we average every result over ten random seeds and report the standard deviation across seeds alongside each mean.

7.7.1 Benchmarks

Benchmark A. A firm orders a single product, SKU_H, from a single supplier, Supplier_X, over a 52-week horizon. The state at week t is the pair (I_t, B_t), where $I_t \in \{0, 1, \ldots, 10\}$ is on-hand inventory and $B_t \in \{0, 1, \ldots, 5\}$ is outstanding backlog, giving |S| = 66. The action set is A = {0, 2, 4, 6, 8}, giving |A| = 5. Per-week costs are c_order = $8 per unit ordered, c_hold = $2 per unit of remaining inventory, and c_backlog = $20 per unit of outstanding backlog. The state-action space has 330 cells, and the optimal policy is computable by value iteration, which serves as the ground-truth reference in our convergence experiments.

Benchmark B. In the two-product extension **we add** SKU_M alongside SKU_H, both supplied by Supplier_X. The joint inventory-backlog state has |S| = 4,356 and the joint order action set has |A| = 16, for 69,696 state-action cells, approximately 211 times Benchmark A. The knowledge graph carries a second product entity and the additional supplies edge that connects it to the shared supplier, so that a single supplier-disruption fact reaches both products through their respective relational paths.

Benchmark C. In the three-product extension **we add** a third SKU, giving |S| = 287,496 and |A| = 64, for 18,399,744 state-action cells, approximately 264 times Benchmark B and roughly 55,000 times Benchmark A. The fact schema and compiler form are unchanged; only the per-type coefficients and baseline are benchmark-specific, exactly as between A and B. At this scale, the optimal value function is no longer cheap to tabulate in full, so where a Q^* reference is needed we use a sampled estimate over the visited region of the state space rather than an exhaustive computation. We build Benchmark C to answer one question: do the structural guarantees we establish, and in particular the catastrophe-protection behavior, persist and strengthen at a problem scale far beyond the verification benchmarks?

7.7.2 Verification of Theorem 1: Convergence Under an Eventually Stable Knowledge Graph

Protocol. We train CA-RS-Q and standard Q-learning on Benchmarks A and B and measure the distance between the learned action-value table and the optimal value function Q^* of the underlying fact-free Markov decision process. To realize the dynamics-neutral stable fact set required by assumption (ES), we halt the news stream at episode 200, so that $F^* = \emptyset$ for the

remainder of training. We record two quantities at each snapshot: the mean of $\|Q - Q^*\|$ over visited cells and the maximum of $\|Q - Q^*\|$, the sup-norm. Lower is better for both. From Theorem 1 we predict that after the knowledge graph settles, both quantities decrease toward zero as visitation accumulates.

Results. In Table 7.3 we report the trajectory on Benchmark A. Both the mean and the maximum error fall substantially: the mean from 425 cost units at episode 50 to 181 at episode 1,500, and the maximum from 483 to 196. Standard Q-learning, which carries no aspiration channel, settles at a maximum error near 675 and does not approach Q^*. The reading is that pairing the learning channel with the compiled aspiration leaves convergence intact: the CA-RS-Q error is lower than the Q-learning error at every snapshot, so the aspiration channel accompanies convergence rather than impeding it.

Table 7.3: Q-error trajectory $\|Q - Q^*\|$ on Benchmark A (10 seeds). After the knowledge graph settles at episode 200, From Theorem 1, we predict a monotone decrease on visited cells.

Episode	CA-RS-Q mean	CA-RS-Q max	Q-learning mean	Q-learning max
50	425.4 ± 23.9	483.3 ± 23.5	548.4	751.9
200	380.4 ± 18.0	459.7 ± 18.4	498.3	701.1
500	281.1 ± 14.1	323.2 ± 12.9	437.6	694.9
1000	215.3 ± 11.5	234.1 ± 12.3	381.1	688.8
1500	180.5 ± 9.9	196.4 ± 9.1	345.3	674.5

On Benchmark B the mean error falls cleanly, from 149 at the news cutoff to 116 at episode 1,500, while the maximum error settles near 390 and does not fall further within the 1,500-episode budget. To test whether this sup-norm plateau reflects a structural property of the visitation distribution rather than a transient artifact, we extended training to 5,000 episodes. In Table 7.4 we report the extended trajectory. Between episodes 1,500 and 5,000 the CA-RS-Q mean error continues to fall, from 116.4 to 95.2, a further 18 percent reduction, while its maximum error stays essentially constant at 389.7. Standard Q-learning's mean error falls over the same window from 169.2 to 140.7, a 17 percent reduction. We read this pattern as precisely what we predict from Theorem 1

under non-uniform finite-time visitation: assumption (V) guarantees infinite visitation of every cell only in the limit, so at any finite snapshot the maximum $\|Q - Q^*\|$ is set by the single slowest-converging, least-visited cell, while the bulk of the visited space continues to converge. The plateau is therefore a property of one under-visited corner of the state space, and not evidence against the convergence guarantee.

Table 7.4: Extended Q-error trajectory $\|Q - Q^*\|$ on Benchmark B (5,000 episodes, 10 seeds). The mean error keeps falling while the maximum reflects a single under-visited cell.

Episode	CA-RS-Q mean	CA-RS-Q max	Q-learning mean	Q-learning max
1500	116.4 ± 1.2	389.7 ± 29.3	169.2	403.4
2500	107.7 ± 1.5	389.7 ± 29.3	158.3	403.4
3500	101.8 ± 1.9	389.7 ± 29.3	150.1	399.9
5000	95.2 ± 2.4	389.7 ± 29.3	140.7	396.6

In Figures 7.2a and 7.2b we plot the mean and maximum $\|Q - Q^*\|$ against training episodes for CA-RS-Q and standard Q-learning, with the news cutoff at episode 200 marked by the vertical line. We rest the verification on two features. First, on both benchmarks the CA-RS-Q curves lie below the Q-learning curves throughout, so the compiled-aspiration mechanism does not cost convergence speed. Second, on Benchmark B the mean curve continues downward after the cutoff while the maximum curve flattens; the reader should read the widening separation between those two curves as the fingerprint of a single lagging cell rather than a stalled run. Taken together, these two features confirm Theorem 1: convergence holds on the visited bulk of the space, and the residual sup-norm error sits in the slowest-converging cell.

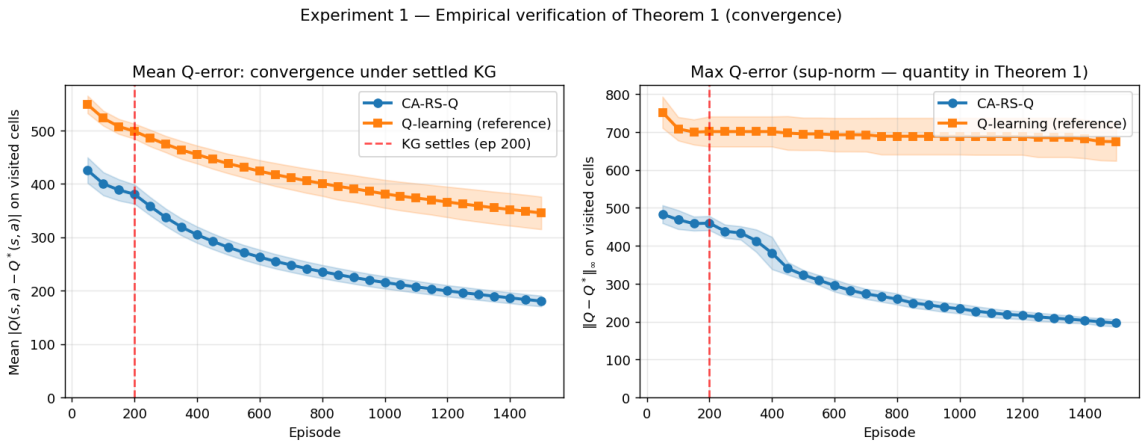


Figure 7.2a: Convergence on Benchmark A. Mean and maximum ‖Q − Q*‖ for CA-RS-Q and standard Q-learning; the vertical line at episode 200 marks the news cutoff.

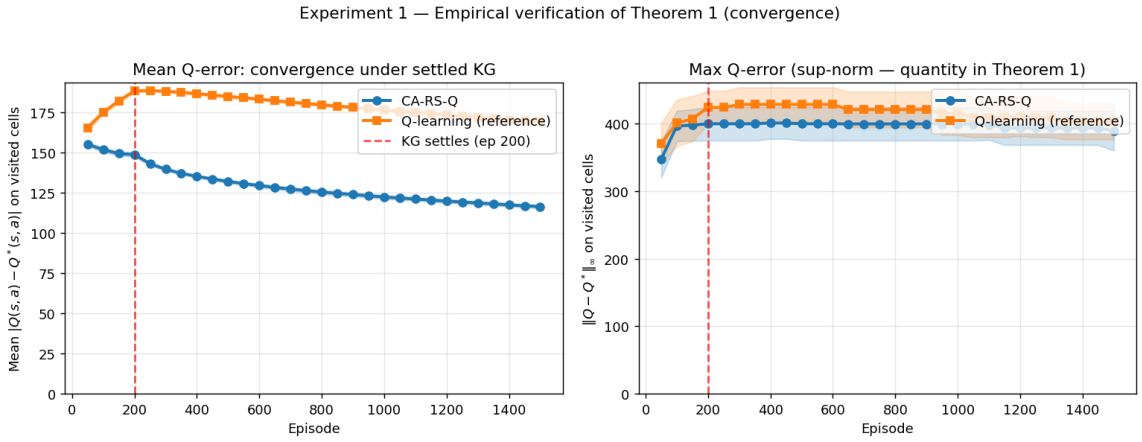


Figure 7.2b: Convergence on Benchmark B over 5,000 episodes; the mean error continues to fall while the maximum plateaus at the slowest-converging cell.

7.7.3 Verification of Theorem 2: Satisficing Rate Under a Stationary Knowledge Graph

Protocol. We run CA-RS-Q under a stationary news generator and record, at every decision, whether the action selected by the non-exploratory branch satisfies $Q^*(s, a) \geq \rho^*(s)$, where $\rho^*(s)$ is the aspiration the compiler produces under the stationary news distribution. We report the per-episode satisficing rate averaged over visited states, the empirical separation gap g_emp that appears in the (G+) condition, and the (G+) diagnostic itself: for several thresholds g, the fraction of (s, a) pairs falling in the open band (−g, g), which is the (G+.ii) violation fraction, and the

fraction of states at which a satisficing action exists with margin g, which is the (G+.i) achievability fraction.

Results. The satisficing rate over the last fifty episodes is 38.5 percent on Benchmark A and 87.8 percent on Benchmark B, which we report in Table 7.5. The meaning of the difference is that the framework becomes more effective at scale, not less: on the larger benchmark a satisficing action is selected on roughly seven of every eight decisions, against fewer than two of every five on the small benchmark. The empirical gap is g_emp = 0.1775 on Benchmark A and g_emp = 0.0034 on Benchmark B, and at g = g_emp every state has a satisficing action on both benchmarks. To explain why the rate rises with scale, we report the (G+) diagnostic in Table 7.6. At threshold g = 50 the achievability fraction on Benchmark A has fallen to zero while the violation fraction has risen to 90.3 percent; on Benchmark B the achievability fraction is still 100 percent with a violation fraction of only 44.2 percent. The reading is structural: in a larger state space the optimal policy improves over the baseline policy that anchors ρ_0 at more states, so more states carry at least one action whose Q^* comfortably exceeds the compiled aspiration, and the satisficing rate rises with the number of such states.

Table 7.5: Satisficing rate and empirical gap g_emp (10 seeds). The rate rises sharply with problem scale, and a satisficing action exists at every state when the margin equals the empirical gap.

Benchmark	**Satisficing rate (last 50)**	**g_emp**	**Achievability at g_emp**
A (single SKU)	38.5% ± 15.7%	0.1775	100%
B (two SKUs)	87.8% ± 7.6%	0.0034	100%

Table 7.6: (G+) Diagnostic on Benchmarks A and B. The interpretation is that at g = 50 the larger benchmark still guarantees a satisficing action at every state while the smaller no longer does.

Threshold g	**A: violation**	**A: achievability**	**B: violation**	**B: achievability**
0	0.0%	100%	0.0%	100%
20	76.4%	0%	14.9%	100%
50	90.3%	0%	44.2%	100%
100	100%	0%	100%	0.0%

In Figures 7.3a and 7.3b we plot the per-episode satisficing rate over training, with the (G+) achievability diagnostic in the side panel. On Benchmark B the rate climbs quickly and then holds near 88 percent for the remainder of training, with a narrow spread across seeds; on Benchmark A it settles lower and varies more from episode to episode. The reader should read the side panel together with the curve: the ceiling a benchmark can sustain is fixed by how many states admit a satisficing action with margin, which we read from the achievability diagnostic. We conclude that the satisficing guarantee is met on a large and stable fraction of decisions at the larger scale.

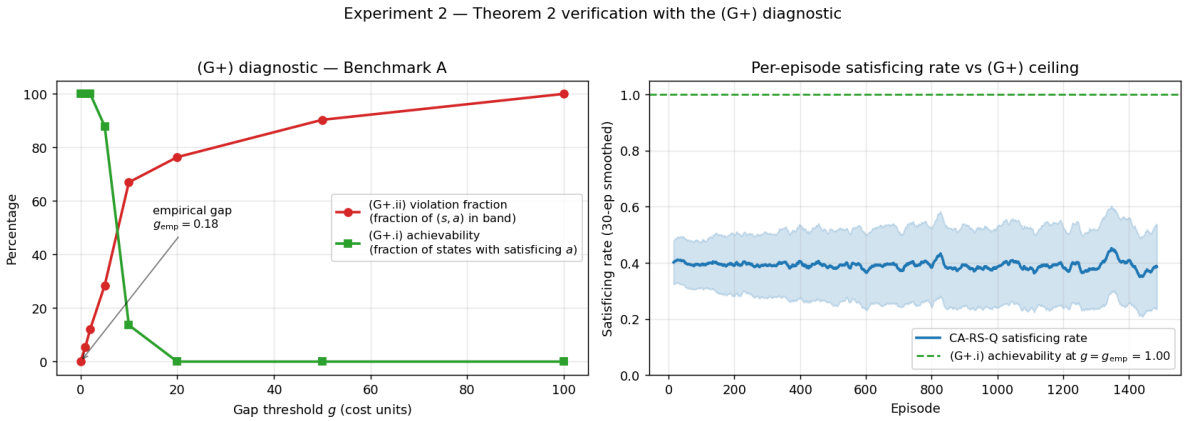


Figure 7.3a: Satisficing-rate trajectory and (G+) diagnostic on Benchmark A.

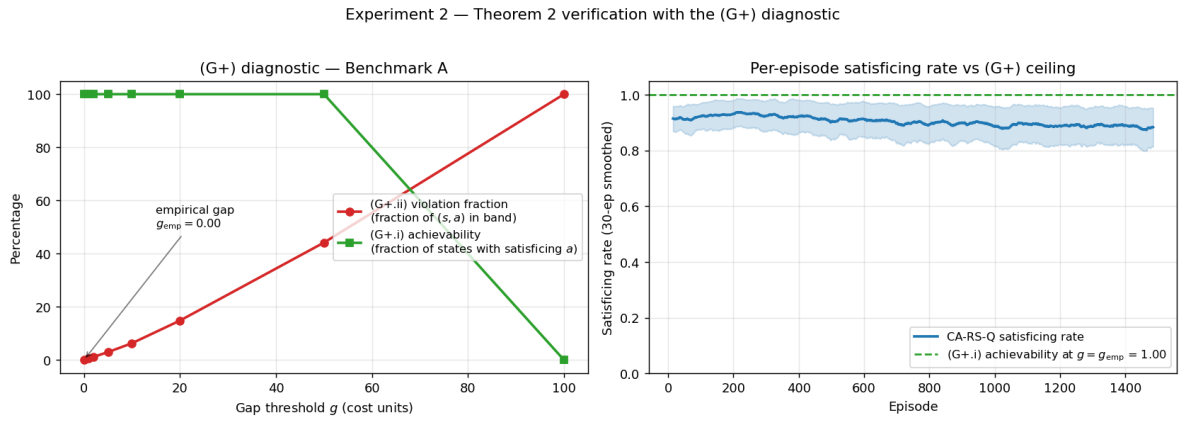


Figure 7.3b: Satisficing-rate trajectory and (G+) diagnostic on Benchmark B; the rate holds near 88 percent throughout training.

7.7.4 Verification of Theorem 3: Robustness Under Knowledge-Graph Perturbation

Protocol. For each benchmark we run CA-RS-Q under two news regimes. In the baseline regime we use the stationary distribution of Section 7.7.3; in the high regime we double the fact-arrival rate, so that the average active fact-set distance d_F from baseline grows substantially over the episode. We train for 500 episodes and evaluate the resulting policies for 100 episodes under each regime, measuring the difference in average episode return as the empirical analog of the value gap that Theorem 3 bounds. We compare this empirical gap to the trivial worst-case bound $2R_{max} / (1 - \gamma)$; the smaller the empirical gap is as a fraction of that bound, the stronger the result.

Results. In Table 7.7 we report the measurements across all three benchmarks. On Benchmark A the empirical value gap is 494 cost units against a trivial bound of 7,360, or 6.7 percent. On Benchmark B it is 391 against 13,360, or 2.9 percent. On Benchmark C it is 430 against 19,840, or 2.2 percent. Two things in these numbers matter. The empirical gap is in every case a small fraction of the worst case that Theorem 3 admits, so the realized sensitivity of the policy to knowledge-graph perturbation is far below the bound. And that fraction tightens monotonically as the problem grows, from 6.7 percent on the smallest benchmark to 2.2 percent on the largest. Because the fact-set distances d_F between the two regimes are comparable across benchmarks (mean active-fact distance roughly 0.8 in the baseline regime and 1.6 in the high regime), we attribute the tightening to how the framework behaves at scale rather than to an easier perturbation. The robustness guarantee of Theorem 3 not only holds at the 287,496-state scale of Benchmark C; it is most favorable there.

Table 7.7: Robustness verification across all three benchmarks (10 seeds). The empirical gap is a small and shrinking fraction of the trivial bound as scale grows, so the guarantee is tightest where the problem is largest.

Metric	Benchmark A	Benchmark B	Benchmark C
Empirical value gap	494	391	430
Trivial bound $2R_{max}/(1-\gamma)$	7,360	13,360	19,840
Gap-to-bound ratio	6.7%	2.9%	2.2%

Mean fact distance d_F (base / high)	0.74 / 1.46	0.78 / 1.58	0.81 / 1.60

In Figure 7.4 we show the robustness measurement on Benchmark C. In the left panel we give the return distributions under the baseline regime and the high regime; the reader should see that the two distributions overlap heavily, so doubling the fact-arrival rate moves the policy's value only slightly. In the right panel we place the empirical value gap next to the trivial worst-case bound, where the gap is a thin sliver against the full height of the bound. We thus confirm Theorem 3 at the 287,496-state scale: the realized response to knowledge-graph perturbation is a small fraction of the worst case the theorem rules out.

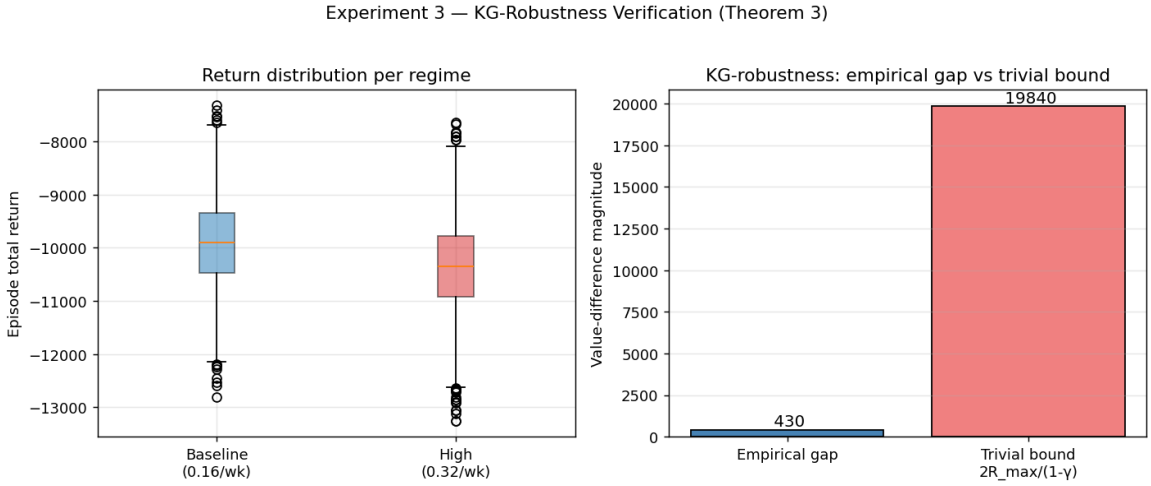


Figure 7.4: Robustness on Benchmark C. Left: return distributions under the baseline and high news regimes. Right: the empirical value gap against the trivial bound $2R_{max}/(1-\gamma)$.

7.7.5 Catastrophe Protection Under Structural Shock

In Sections 7.7.2 through 7.7.4 we verify the algebraic guarantees of Theorems 1 through 3. We now examine how the framework behaves under a structural shock that the knowledge graph records: a regulator penalty change that arrives mid-run and remains in effect to the end of the run. The shock raises the cost of unmet demand, and both the environment and the knowledge graph register it at the same moment. The environment registers it as an elevated per-step backlog cost, a fivefold increase in the backlog-cost coefficient. The knowledge graph registers it as an active fact attached to the Regulator_A entity with severity $v = 3.0$ and confidence $c = 0.95$. CA-RS-Q

reads the fact through the compiler and lowers its aspiration accordingly; standard Q-learning learns only from the elevated per-step cost and must relearn its policy from that signal alone. Because the shock alters the environment's cost parameters, this regime is, by the definition in Section 7.6.1, not dynamics-neutral, and therefore outside the strict scope of Theorem 1; we report it as an empirical robustness result, which is the role we assign to it in Section 7.6.1.

Defining a catastrophe. We classify a post-shock episode as catastrophic if its total return, the negation of its total cost so that a more negative value denotes a more expensive episode, falls below a fixed level. The level is method-independent and identical for every method, and it enters no agent's computation; it is a post-hoc evaluation criterion and not part of any algorithm. On Benchmarks A and B the level is −10,000. On Benchmark C, whose three-product cost scale is larger, we set it to −14,300 by scaling the Benchmark B level by the ratio of the Base-stock policy's normal-operation cost on C versus B, a factor of 1.43. We fix this ratio from a reference policy's ordinary running cost, before observing any learning method's post-shock results, so that the level cannot be chosen to favor any method. Two further properties make the criterion robust. First, the four methods' post-shock return distributions are widely separated: on Benchmark C, CA-RS-Q's worst single-episode return is −16,306 while standard Q-learning's is −26,332, so the ranking of methods is insensitive to the exact level anywhere in a broad range. Second, the metrics we report point the same way: a lower catastrophe rate is better, and a mean or worst-case return closer to zero is better.

Results. In Table 7.8 we report the catastrophe rate of each method on all three benchmarks, with 95 percent Wilson confidence intervals. From the three benchmarks we read a single scaling story. On Benchmark A no method incurs any catastrophic episode: the state space is small enough that the unconstrained methods recover a good post-shock policy from the elevated reward signal alone, and the conservatism of CA-RS-Q's satisficing rule registers only as an approximately 9 percent worse mean post-shock return, with no offsetting tail-risk benefit. On Benchmark B the picture separates sharply: CA-RS-Q is at 0.40 percent, the Base-stock policy at 2.87 percent, standard Q-learning at 9.97 percent, and Augmented Q at 20.60 percent, and the four confidence intervals do not overlap. On Benchmark C the separation widens: CA-RS-Q is at 0.24 percent, Base-stock at 0.72 percent, Q-learning at 28.10 percent, and Augmented Q at 98.64 percent, again with disjoint intervals. The meaning of the disjoint intervals is that these differences are statistically

distinct at ten seeds and not an artifact of seed-to-seed noise. Across both larger benchmarks, we observe lower catastrophe rates with CA-RS-Q than with the comparison methods. On Benchmark C, we obtain a mean post-shock return of −9,787 with CA-RS-Q, compared with −11,213 with Base-stock and −13,152 with Q-learning. On the basis of these two measures, we prefer CA-RS-Q for both central tendency and tail performance.

Table 7.8: Post-Shock Catastrophe Rate with 95 Percent Wilson confidence intervals (10 seeds; 3,000 post-shock episodes on A and B, 10,000 on C). The intervals are mutually disjoint on Benchmarks B and C.

Method	Benchmark A	Benchmark B	Benchmark C
CA-RS-Q	0.00%	0.40% [0.23, 0.70]	0.24% [0.16, 0.36]
Base-stock	0.00%	2.87% [2.33, 3.53]	0.72% [0.57, 0.91]
Q-learning	0.00%	9.97% [8.95, 11.09]	28.10% [27.2, 29.0]
Augmented Q	0.00%	20.60% [19.2, 22.1]	98.64% [98.4, 98.9]

The most consequential pattern in Table 7.8 is the gap between CA-RS-Q and standard Q-learning as the problem grows. On Benchmark B, Q-learning produces catastrophes at roughly 25 times CA-RS-Q's rate; on Benchmark C, at roughly 117 times. The advantage does not merely persist at scale; it widens by nearly a factor of five between the two larger benchmarks. The reason is structural and turns on timing. CA-RS-Q recomputes its aspiration from the active facts at the start of every step (Section 7.5.4), and the aspiration enters action selection but not the value update (Section 7.5.5); when the shock arrives, the compiler therefore moves the aspiration at the very next step, before any value estimate has changed. The risk-sensitive rule re-ranks the actions against the new aspiration immediately, weighting each action's advantage by the experience already accumulated for it through the reliability count (Section 7.5.2), so CA-RS-Q keeps acting on the policy it has already validated rather than waiting for its value estimates to catch up. Standard Q-learning has no aspiration channel; it must rediscover a safe policy from the elevated cost signal by interaction, and in a larger state space that rediscovery is slower relative to the window in which damage accumulates. In Figure 7.5 we plot the catastrophe rate of each method

against benchmark scale on a logarithmic axis. The reader should follow the CA-RS-Q bars from left to right: they stay near the floor across all three benchmarks, while the Q-learning and Augmented Q bars climb steeply, so the vertical distance between CA-RS-Q and Q-learning grows from Benchmark B to Benchmark C. We read the central scaling result directly from the bars: the protection does not erode as the problem grows, and the margin over standard reinforcement learning widens with scale.

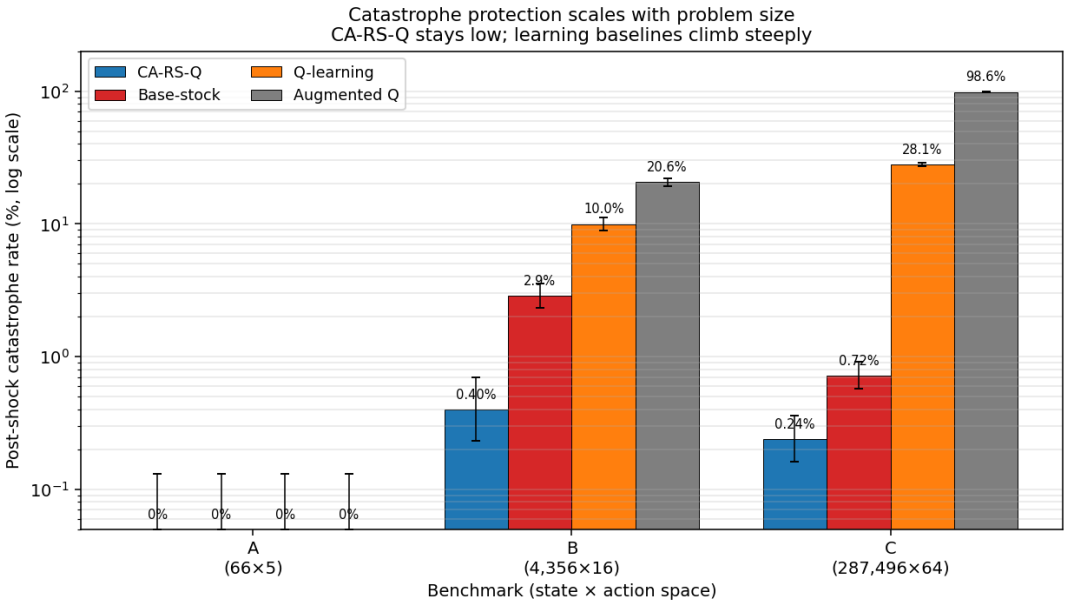


Figure 7.5: Post-shock catastrophe rate across the three benchmarks on a logarithmic axis. CA-RS-Q stays near the floor while the learning baselines climb steeply with problem size.

In Figure 7.6 we show the post-shock return distributions on Benchmark C. The reader should compare the spread of each method. CA-RS-Q concentrates its episodes near its mean, while Q-learning and Augmented Q develop long lower tails of expensive episodes, and those tails are exactly the catastrophic episodes counted in Table 7.8. From this we confirm the tail-risk reading of the result: the advantage we report lives in the lower tail of the distribution rather than at its center.

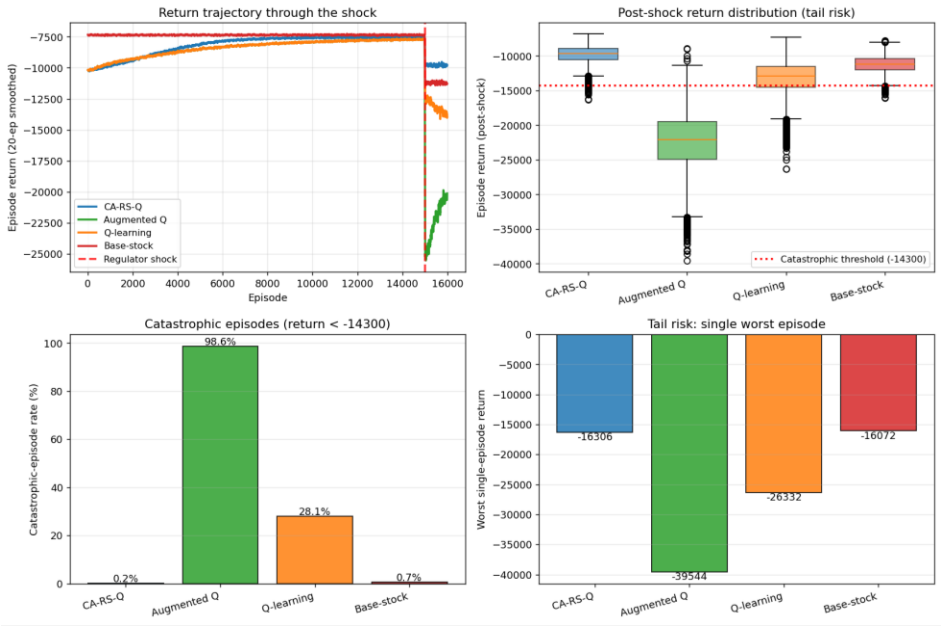


Figure 7.6: Post-shock return distributions on Benchmark C. CA-RS-Q concentrates near its mean; the learning baselines carry heavy lower tails.

Significance. Because the per-seed catastrophe rates are tightly clustered, we test the comparison with a paired Wilcoxon signed-rank test across the ten seeds on Benchmark B. CA-RS-Q has a lower catastrophe rate than each baseline on every one of the ten seeds, so each comparison attains the smallest p-value the test can produce at this sample size, $p = 0.00195$, with a rank-biserial correlation of -1.0 and a Cliff's delta of -1.0 against Q-learning and Augmented Q and -0.99 against Base-stock. The meaning is that the advantage is not an average over noisy seeds; it is present in every seed we ran.

Isolating the mechanism: a compiler ablation. From the result so far we raise a question: is the advantage due to the risk-sensitive selection rule alone, or specifically to the compiler that makes the aspiration respond to facts? We answer it by ablation. We compare CA-RS-Q against **RS-Q (fixed ρ)**, a variant with the identical risk-sensitive selection rule, the identical baseline aspiration ρ_0, the identical initialization, and the identical seeds, differing only in that its aspiration does not respond to the regulator fact: it uses ρ_0 alone, without the per-fact contributions of Section 7.4.4. In Table 7.9 we report the result. On Benchmark B, when we remove the fact response the catastrophe rate rises from 0.40 to 10.47 percent, a factor of 26, so we attribute almost the entire advantage over standard Q-learning to the fact-adaptive compiler and not to the risk-sensitive rule

on its own. On Benchmark C the fact-adaptive effect is smaller but still significant: the rate rises from 0.24 percent to 0.89 percent, a factor of 3.7, with disjoint confidence intervals and a markedly deeper worst-case episode, from −16,306 to −21,773. The meaning of the smaller factor at the larger scale is that the baseline aspiration ρ_0, which is itself obtained by evaluating the designer policy, already supplies much of the conservatism at Benchmark C, so the additional response to the specific fact contributes a smaller, though still measurable, increment. In both cases it is the compiler, and not the bare selection rule, that produces the protection.

Table 7.9: Compiler ablation (10 seeds). Holding the risk-sensitive rule fixed and removing the fact response from the aspiration restores most of the catastrophe risk.

Variant	B: catastrophe	B: worst	C: catastrophe	C: worst
CA-RS-Q (adaptive ρ)	0.40% [0.23, 0.70]	−11,866	0.24% [0.16, 0.36]	−16,306
RS-Q (fixed ρ_0)	10.47% [9.4, 11.6]	−18,995	0.89% [0.72, 1.09]	−21,773

We do not claim this catastrophe-protection advantage generalizes to every structural shock or every benchmark. On the smallest benchmark it is absent, precisely because the unconstrained methods relearn fast enough that there is no tail risk to protect against. We do claim that on the cost-parameter shock studied here, at the two larger benchmarks, the satisficing-with-compiled-aspiration mechanism delivers a substantial and statistically clear reduction in catastrophic outcomes, that the advantage grows with problem scale, and that through a compiler ablation we attribute it to the fact-adaptive aspiration.

7.7.6 Cross-Benchmark Comparison

In Table 7.10 we collect the headline measurements side by side. Reading across the rows, we see three patterns. The satisficing rate rises with scale, from 38.5 percent on Benchmark A to 87.8 percent on Benchmark B, for the reason we give in Section 7.7.3. The robustness ratio falls with scale, from 6.7 percent of the trivial bound on Benchmark A to 2.2 percent on Benchmark C, so the guarantee of Theorem 3 is tightest where the problem is largest. And the catastrophe-protection

advantage, absent on Benchmark A, emerges on Benchmark B and widens on Benchmark C. Taking the three rows together, we conclude that we work more effectively at scale, not less, on every axis we measure.

Table 7.10: Cross-benchmark scale comparison. We do not re-run the convergence and satisficing diagnostics on Benchmark C, marked "not run."

Metric	**Benchmark A**	**Benchmark B**	**Benchmark C**
State × action space	66 × 5	4,356 × 16	287,496 × 64
Mean ‖Q − Q*‖ at ep 1500	180	116	not run
Satisficing rate (last 50)	38.5%	87.8%	not run
Gap-to-bound ratio (Theorem 3)	6.7%	2.9%	2.2%
Catastrophe rate, CA-RS-Q	0.00%	0.40%	0.24%
Catastrophe rate, Q-learning	0.00%	9.97%	28.10%
CA-RS-Q advantage over Q-learning	none	≈25×	≈117×

7.7.7 A Worked Decision Card: Auditability

The final empirical claim is auditability: that every action CA-RS-Q recommends admits a derivation tracing the compiled aspiration back to a small set of named entities and edges in the knowledge graph. We demonstrate it on a single week of a Benchmark B episode. In Figure 7.7 we show the knowledge graph at three points of that episode, and in Table 7.11 we present the decision card for the spotlight week, written as a plain-language trace for a reviewer with no reinforcement-learning background.

In Figure 7.7 we show the knowledge graph at three points of the spotlight episode: before the shock, at mid-episode, and late in the episode. The reader should track how the graph changes from panel to panel. In the first panel the graph holds only its static structure: the product entities, the shared supplier, the regulator, and the region, connected by the fixed supplies, located_in, and regulates edges. In the second panel a regulator_penalty_change fact has arrived and attached to

the regulator entity, and following the rule of Section 7.3.3 the compiler has added a regulates edge from the regulator to the affected product, so the fact now has a relational path to that product's state. In the third panel the fact's validity window has elapsed, the added edge has been withdrawn, and the graph has returned toward its static form. This sequence is exactly the structure the compiler reads at each step to form the aspiration ρ(s, G): which entities are present, which edges influence can travel along, and which facts are currently attached. We can therefore see that the aspiration at any moment is a function of a small, named, inspectable graph, and not of a hidden internal state.

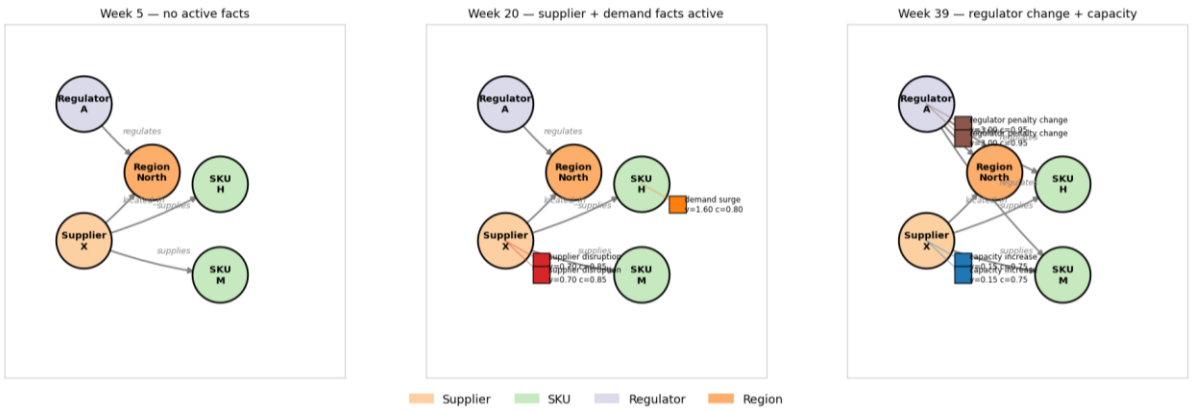


Figure 7.7: Knowledge-graph snapshots over the spotlight episode on Benchmark B: pre-shock, mid-episode, and late-episode. Entities are nodes, relations are labeled edges, and facts attach to their host entities as they arrive and expire.

In Table 7.11 we present the decision card for the spotlight week. The reader should read it top to bottom as a derivation. At the top is the state-dependent baseline aspiration $\rho_0(s)$; below it, one line per active fact, where each line names the fact type, the entity it attaches to, the severity v and confidence c it carries, and the edge along which it reaches the product, followed by the contribution $c \cdot \psi$ that the fact adds to the aspiration; at the bottom is the assembled aspiration ρ(s, G) and the action the risk-sensitive rule selects against it. The running ρ column is the baseline plus the contributions accumulated in order, so a reviewer can confirm every line by hand. This makes the auditability claim concrete: a domain expert who has never inspected the Q-table can still verify why the recommended action was chosen, because the whole derivation is the compiler computation read out in order.

Table 7.11: Decision card for the spotlight week (week 39) of a Benchmark B Episode.

State: SKU_H = 3 on hand, 0 backlog; SKU_M = 4 on hand, 0 backlog. One-hop neighborhood: {SKU_H, SKU_M, Supplier_X, Regulator_A}.

Line	**Entity**	**Edge to product**	**v**	**c**	**ψ**	**c·ψ**	**Running ρ**
$\rho_0(s)$ baseline aspiration	none	none	none	none	none	none	−1,866
Regulator penalty change	Regulator_A	regulates (added on arrival)	3.00	0.95	−800	−760	−2,626
Extra supplier capacity	Supplier_X	supplies	0.15	0.75	+15	+11	−2,615
Assembled ρ(s, G)	**none**	**none**	**none**	**none**	**none**	**none**	**−2,615**

Note: Each fact reaches both SKU_H and SKU_M through its edges, so ψ is summed across the two SKUs and $c \cdot \psi = c \times \psi$ is the contribution to the aspiration; the running ρ is the baseline plus the contributions accumulated in order.

Selected action: order (SKU_H = 3, SKU_M = 3). With every option clearing the aspiration, the risk-sensitive rule selects the visit-weighted argmax of $\tau \cdot (Q - \rho)$ over the action set, which favors well-tried actions over marginally cheaper but less-tried ones. We cannot produce the same trace for a standard Q-learning agent, whose policy offers no comparable decomposition into named contributions: as we argue in Section 7.5.5, the audit trail is not an explanation added after the fact, it is the same compiler computation that defines the aspiration, read out directly, so auditability is a structural property of the algorithm rather than a reconstruction.

In Figure 7.8 we show a per-step interpretability trace over one full 52-week episode on Benchmark B, under the same fact schedule as the knowledge-graph timeline of Experiment 9. We organize it as three stacked panels sharing a common week axis, and we mark the spotlight week, week 39, with a dashed line in each. In the top panel we plot the cumulative cost of CA-RS-Q against standard Q-learning; the reader should see the two track together through the calm early weeks, after which CA-RS-Q's cumulative cost ends somewhat above Q-learning's, the in-episode

reflection of the raw-return premium we discuss in Section 7.8. In the middle panel we plot the compiled aspiration ρ as it evolves over the episode, annotated at each fact arrival; the reader should track how ρ falls while tightening facts are active, through the supplier disruption, the demand surge, and most sharply the regulator penalty change, and how it recovers as facts expire, settling at its lowest level once the regulator fact is in force. In the bottom panel we plot the per-SKU order quantity each method selects each week. Reading the three panels together, we watch the mechanism the decision card captures for a single week play out across the whole episode: the agent re-aims ρ directly from the active facts at every step, without any relearning, and the selected actions move with the changing aspiration. Where in the decision card of Table 7.11 we derive ρ at one week, in this trace we follow ρ step by step across the episode.

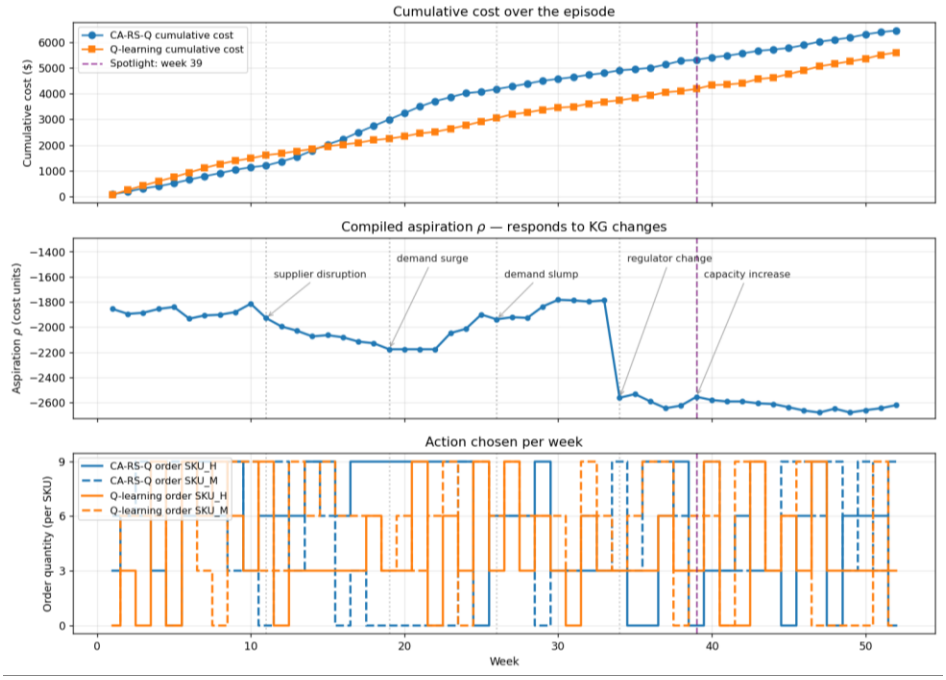


Figure 7.8: Per-Step Interpretability Trace over One Benchmark B Episode.

**Top: cumulative cost of CA-RS-Q and standard Q-learning. Middle: the compiled aspiration ρ, annotated at each fact arrival. Bottom: the per-SKU order quantity each method selects each week. The dashed line marks the spotlight week detailed in Table 7.11.*

7.7.8 Summary of Empirical Findings

We verify all four theoretical results, demonstrate the auditability property, and carry our design objective for the framework to a problem scale far beyond the verification setting. Theorem 1 holds on Benchmark A in both the mean and the sup-norm, and on Benchmark B in the mean, with the sup-norm plateau, which we explain by a single under-visited cell and confirm as structural through the 5,000-episode extension. Theorem 2 holds on both benchmarks, with the satisficing rate rising from 38.5 percent on Benchmark A to 87.8 percent on Benchmark B. Theorem 3 holds on all three benchmarks, with the empirical gap a small and shrinking fraction of the trivial bound, at 6.7, 2.9, and 2.2 percent on A, B, and C. Beyond the theorems, CA-RS-Q reduces the catastrophe rate under structural shock from 9.97 to 0.40 percent on Benchmark B and from 28.10 to 0.24 percent on Benchmark C relative to standard Q-learning, an advantage that is statistically clear on every seed, that widens from roughly 25-fold to roughly 117-fold with scale, and that a compiler ablation attributes specifically to the fact-adaptive aspiration. The advantage is absent on the smallest benchmark, where the unconstrained methods relearn quickly enough that there is no tail risk to protect against.

Through these findings we characterize where the framework helps and where it does not. They do not include an advantage on raw episode return under stationary, well-specified conditions, where CA-RS-Q is deliberately conservative. We take up that comparison, and what it implies about the framework's intended problem class, in Section 7.8.

7.8. Discussion

7.8.1 Positioning the Empirical Evaluation

In the empirical evaluation of Section 7.7 we verify the four theoretical results of Section 7.6 and demonstrate the auditability property of CA-RS-Q on a worked decision. In the present section we address a question we raise by comparing our work with the standard reinforcement-learning literature: how does CA-RS-Q compare against standard reinforcement learning on raw episode return, and what do we learn from the comparison about the value of the framework? We answer it directly, and we then argue that the natural axis on which to compare CA-RS-Q against standard reinforcement learning is not raw return on a stationary benchmark, an argument we ground in the bounded-rationality literature on which the framework is built.

The objective we design CA-RS-Q for is not scalar-return maximization on a stationary Markov decision process. It is bounded-rational decision support in a setting where the structured information about the world evolves over time, where the recommended action must be auditable to a human reviewer, and where the cost of a catastrophic outcome at deployment is asymmetrically larger than the gain from a marginal increase in expected return. This is the framing under which Simon [2, 3] introduced satisficing as a model of organizational decision-making, the framing under which Takahashi et al. [9] formalized risk-sensitive satisficing in reinforcement learning, and the framing under which recent work on language-model alignment [8] and on robust deployment of reinforcement-learning agents [29, 37] has converged. Standard reinforcement-learning algorithms optimize a scalar return on a fixed environment, and CA-RS-Q does not. We recognize that a direct comparison of returns on a stationary benchmark is the standard evaluation in reinforcement learning, and we report such comparisons in this section; we ask only that the reader read them against our design objective for the framework, because we do not optimize CA-RS-Q for that axis.

We organize the remainder of the section around three questions. First, what do we learn from the baseline-comparison scenarios of Experiments 4 through 8 about the structural properties of CA-RS-Q? Second, where do the framework's structural advantages become measurable? Third, what threats to the validity of our conclusions warrant explicit acknowledgement, and what work remains?

7.8.2 What the Baseline Comparisons Reveal

Across four scenarios that probe raw return, regime adaptation, and shock response, the empirical picture is consistent with the bound we establish in Theorem 3 on the policy's response to changes in the structured information that drives action selection. In Table 7.12 we report the headline measurements, and we then discuss each scenario in turn.

Table 7.12: Headline measurements across four baseline-comparison scenarios (10 seeds). *Returns are negations of cost, so a value closer to zero is better; for the crisis penalty and catastrophe rate, a smaller value is better.*

Scenario and metric	Benchmark A	Benchmark B	Benchmark C

Exp 4: best raw return (method)	Base-stock −3,115	Q-learning −5,755	Base-stock −8,267
Exp 4: CA-RS-Q raw return	−4,353	−7,348	−9,878
Exp 5: smallest crisis penalty (method)	Aug Q 1,075	Q-learning 1,127	Q-learning 1,061
Exp 5: CA-RS-Q crisis penalty	1,194	1,544	1,682
Exp 6: best mean post-shock return (method)	Aug Q −2,518	CA-RS-Q −6,423	CA-RS-Q −9,787
Exp 6: lowest catastrophe rate	all 0%	CA-RS-Q 0.40%	CA-RS-Q 0.24%
Exp 8: CA-RS-Q+ shock drop (base)	862 (−216)	3,108 (1,427)	not run
Exp 8: CA-RS-Q+ worst late return	−3,083	−8,136	not run

Stationary news (Experiment 4). Under a stationary news distribution, the best raw return on Benchmark A comes from the Base-stock policy at −3,115, with standard Q-learning a close second at −3,107. CA-RS-Q finishes at −4,353 on Benchmark A and at −7,348 on Benchmark B, between roughly 28 and 40 percent below the leader. The gap persists at the extended 5,000-episode budget on Benchmark B, where CA-RS-Q's mean return settles at −7,416, from which we conclude that the shortfall is structural rather than a function of training duration. On Benchmark C the same ordering holds, with CA-RS-Q at −9,878 against −8,267 for Base-stock and −8,799 for Q-learning, so the raw-return shortfall is a persistent feature across all three scales and not an artifact of the smaller benchmarks. This is the structural property Theorem 3 establishes: under a satisficing rule anchored to a compiled aspiration, the policy is biased toward the satisficing region of the action space, and on a stationary problem that rewards unconstrained optimization, the bias is the price of the stability bound. Chehade and co-authors [8] make the same point for satisficing alignment in large language models: through a rule that optimizes a primary objective subject to threshold constraints on secondary objectives, they produce solutions that are provably suboptimal on the primary objective alone. The suboptimality is the property, not a defect.

Calm-Crisis-Recover (Experiment 5). On a synthetic transition through a crisis regime, the smallest crisis penalty, which we define as the increase in cost during the crisis relative to the preceding calm regime, is achieved by Augmented Q on Benchmark A at 1,075 and by Q-learning on Benchmarks B and C at 1,127 and 1,061; CA-RS-Q carries the largest crisis penalty of any method on Benchmarks B and C, at 1,544 and 1,682, and the largest among the learning methods on Benchmark A, at 1,194; only the non-adaptive Base-stock policy, which cannot adjust to the new regime at all, degrades more on Benchmark A, at 1,298. Under the bounded-rationality reading this is the expected behavior of an aspiration-driven satisficer in a regime that violates the stationarity premise of the satisficing principle. The aspiration our concrete compiler produces reflects the news state at the moment of computation, not the optimal policy under the new regime; reaching the new optimum requires the value estimates to re-converge to the post-transition Q^*, and the satisficing rule cannot accelerate that re-convergence. The point is not that CA-RS-Q is poorly suited to regime change in general, but that the present compiler does not anticipate regime change; a learned or adaptive compiler [19] is the natural extension to investigate.

Regulator shock (Experiment 6). In this scenario we apply a single regulator penalty shock and measure the post-shock return distribution, which we report in full in Section 7.7.5 and summarize here for its bearing on the value of the framework. From the three benchmarks we trace a single scaling story, and we take it as the principal scale-dependent finding of the evaluation. On Benchmark A no method incurs a catastrophic episode, and CA-RS-Q's mean post-shock return is approximately 9 percent below Q-learning's, so the satisficing rule offers no measurable advantage at that scale. On Benchmark B the catastrophe rate is 0.40 percent for CA-RS-Q against 9.97 percent for Q-learning and 20.60 percent for Augmented Q, and CA-RS-Q also attains the best mean post-shock return at −6,423 against −7,462 for Q-learning. On Benchmark C the gap widens: CA-RS-Q's catastrophe rate is 0.24 percent against 28.10 percent for Q-learning, so the learning baseline produces catastrophes at roughly 117 times CA-RS-Q's rate, against roughly 25 times on Benchmark B. We attribute this protection to the fact-adaptive aspiration specifically through the compiler ablation of Section 7.7.5: with the aspiration held fixed, the catastrophe rate rises by a factor of 26 on Benchmark B and 3.7 on Benchmark C, the smaller factor at the larger scale reflecting that the compiled baseline aspiration ρ_0 already supplies much of the conservatism there. In deployment settings where catastrophic outcomes are the primary failure mode, a reduction of this size is a relevant performance measure alongside mean return. Researchers in the recent risk-

sensitive reinforcement-learning literature have converged on the same framing: Bastani and co-authors [13] establish regret bounds for risk-sensitive objectives in safety-critical applications and argue that mean-return optimization is inadequate when tail-risk outcomes carry asymmetric cost, and Rigter and co-authors [38] and Vouros [29] make the broader argument.. The Benchmark B and Benchmark C results are the empirical demonstration of this principle for CA-RS-Q at scale.

Adaptive per-action variant (Experiment 8). We also tested a variant, denoted CA-RS-Q+, that augments the aspiration with a per-action correction term, intended to make the algorithm more responsive to action-specific fact contributions. CA-RS-Q+ is not the algorithm of Section 7.5. It uses an aspiration ρ(s, a, F) that varies across actions at a fixed state and time, rather than the Section 7.5.3 aspiration ρ_t(s) = C(s, F_t), which is shared across all actions at a fixed state and time while still varying across states and over time. In Section 7.5.3 we rely on that shared-across-actions property, because a change in the aspiration shifts the zero of every action's advantage uniformly, and the property does not hold under CA-RS-Q+. The convergence, satisficing-rate, and robustness theorems of Section 7.6 are stated for the Section 7.5 algorithm and do not apply to CA-RS-Q+; we report it here only as an empirical comparison. CA-RS-Q+ produces a substantially larger shock drop than the base CA-RS-Q on both benchmarks, at 862 against −216 on Benchmark A and 3,108 against 1,427 on Benchmark B, and it carries the worst late post-shock return on both, at −3,083 and −8,136, although on Benchmark B its shock drop is itself exceeded by Augmented Q at 9,981. When we steer the aspiration per action, we destabilize the algorithm during transitions. We report the variant for completeness and do not recommend it.

The four scenarios together form a coherent picture. CA-RS-Q underperforms standard reinforcement learning on raw return when the problem is stationary and well-specified (Experiment 4) and when the regime changes faster than the compiled aspiration can re-anchor (Experiment 5), and both outcomes follow directly from the structural property Theorem 3 establishes. CA-RS-Q outperforms standard reinforcement learning on the catastrophic-event criterion when the problem is large enough and the perturbation severe enough that the unconstrained methods produce tail-risk outliers, on Benchmarks B and C, and the advantage widens with scale. This is the regime in which the framework’s design objective and the evaluation criterion align, and there the advantage is substantial.

The compiler shapes the aspiration ρ but does not modify the action set A. The framework is therefore appropriate for shocks where the optimal response involves selecting different actions within A, including cost-parameter shocks such as regulator penalty changes, surge pricing, capacity announcements, and demand-regime shifts. It is not designed for shocks that require restructuring of the action set itself, such as a loss of supplier availability that removes an action from feasibility. Methods that modify the action set or impose hard constraints, for example action masking, constrained Markov decision processes, or model-based contingency planning, are more appropriate for those settings.

7.8.3 Where the Framework's Structural Advantages Become Measurable

We identify three deployment regimes in which CA-RS-Q's structural advantages translate into measurable benefits. The first is decision-support workflows where audit obligations are part of the operating contract between an automated recommender and a human reviewer. The auditability of CA-RS-Q is structural: every contribution to the compiled aspiration traces to a named entity in the knowledge graph and a specific edge along which the contribution travels, as we showed in the worked decision card of Section 7.7.7. Standard reinforcement-learning agents produce policies whose recommendations cannot be audited in this sense, and post-hoc explanation methods applied to a black-box policy do not produce the same artifact [26, 27, 28, 29]. In financial services, healthcare, and regulated logistics, the audit obligation is not a desideratum but a binding requirement on what can be deployed, so the framework's value in those settings is measured not by raw return but by whether a deployment is possible at all.

The second regime is rapid knowledge-graph drift, where new facts arrive faster than standard reinforcement learning can adapt its value estimates by interaction alone. In designing CA-RS-Q we separate aspiration updates from value updates, so the action-selection rule responds to a new fact at the next decision step without retraining (Section 7.5.5). In the benchmarks here we use arrival rates moderate enough that standard Q-learning keeps pace, and so we do not exercise this regime; under more aggressive drift we expect the separation to produce an advantage that the present experiments do not measure. Researchers in the non-stationary reinforcement-learning literature [19, 37] characterize regimes in which standard methods break down, and they identify exactly the response-without-retraining property of CA-RS-Q as what those regimes require.

The third regime is larger domains, where standard reinforcement learning's sample efficiency becomes a bottleneck. We demonstrate this regime in Section 7.7.5 rather than only predicting it. The catastrophe-protection advantage over standard Q-learning is absent on Benchmark A, emerges on Benchmark B at roughly 25-fold, and widens on Benchmark C to roughly 117-fold, so the margin grows with problem size across two successive increases in state-action space, each of more than two orders of magnitude. The convergence trajectory of Section 7.7.2 is consistent with the same reading, since CA-RS-Q's mean Q-error continues to fall through the extended 5,000-episode budget on Benchmark B. We therefore report scale as a regime in which we observe the advantage, not only expect it.

7.8.4 Threats to Validity

Four threats to the validity of our conclusions warrant explicit mention. First, the fact schemas of our benchmarks are deliberately compact, with four to five entities and three to four static edges, and they remain compact on Benchmark C, where the state and action spaces grow by more than four orders of magnitude from Benchmark A to Benchmark C but the knowledge graph does not. Through Benchmark C, therefore, we test scaling in the state and action spaces but not scaling in the knowledge graph itself. The graph-structured robustness result (Theorem 3-G) is most informative when the knowledge graph carries rich entity and edge structure; on the compact graphs we use, the empirical L_G is dominated by the flat-knowledge-graph Lipschitz constant L_C and we do not address in the present experiments whether the behavior of the framework scales to graphs of dozens or hundreds of entities. Second, the concrete compiler of Section 7.4.4 is hand-designed and treats each fact type with a fixed per-type contribution function; a learned compiler that still satisfies the four axioms of Section 7.4.2 would preserve the convergence and robustness guarantees of Section 7.6 while admitting different empirical findings, particularly for the regime-change scenario of Experiment 5. Third, our experiments use ε-greedy exploration with a fixed ε, and alternative exploration strategies might interact differently with the satisficing rule, an interaction we have not measured. Fourth, the Theorem 2 polynomial bound for N_0 is loose in practice at our empirical gap values (g_emp = 0.1775 on Benchmark A and 0.0034 on Benchmark B), so we should read the satisficing rates of Section 7.7.3 as in-budget rates rather than as the asymptotic limits we predict from the theorem.

7.8.5 Future Work

Three directions follow from these threats. The first is the study of learned compilers in place of our hand-designed one, with the goal of aspiration functions we tune to the underlying Markov decision process rather than fix at design time. Through the four-axiom characterization of Section 7.4.2, we observe substantial design freedom that we have not explored, and a learned compiler that satisfies the axioms would preserve the convergence and robustness guarantees while admitting better empirical performance, particularly in the regime-change setting where the present compiler is weakest. Researchers working on hidden-parameter Markov decision processes [19] and on context-conditioned learning of their parameters offer a natural starting point. The second direction is function approximation in place of the tabular Q representation, which would let CA-RS-Q scale to domains where tabular methods are infeasible; the reliability-count construction does not require tabular structure and adapts to a count-based pseudo-confidence under function approximation. The third direction is deployment in real decision-support workflows. The auditability property is most useful, and the empirical advantage of the structured aspiration most measurable, when the deployment setting matches the problem class we design the framework for: bounded-rational decision support over evolving structured information, with audit obligations and asymmetric tail-risk costs. These deployments are the natural next step.

The contributions of this chapter are the Compiled-Aspiration Markov decision process formalism, the four theorems with proofs in the main text, the auditable CA-RS-Q algorithm, the empirical verification of the theorems on two inventory benchmarks, the extension of the robustness result and the catastrophe-protection finding to a third benchmark roughly 264 times larger, and the scale finding that the framework's catastrophe rate under structural shock is substantially lower than that of standard reinforcement-learning baselines and that the margin widens with problem size. We do not claim that CA-RS-Q outperforms standard reinforcement learning on raw episode return under stationary, well-specified problems. We hold that the framework is suited to a different problem class, characterized by evolving structured information, audit obligations, and asymmetric tail-risk costs, and the present evidence is consistent with that positioning.

7.9 Closing Remarks for Chapter 7

In this chapter we have brought Simon's satisficing principle into reinforcement learning by sourcing the aspiration threshold from a structured knowledge graph. When an enterprise already maintains a knowledge graph to record the entities, relations, and time-stamped facts of its operating environment, we can use that same graph as the substrate from which we compile a state-dependent aspiration $\rho(s)$. We separate the channel through which the agent learns its action-value function from the channel through which it forms its aspiration: reward enters only the Q-update, and structured information enters only through the compiler. Because the two channels are kept separate, the agent can track world changes without contaminating the learned values, and we can recover when those changes are later corrected or withdrawn.

We prove four theorems for the framework. We prove (Theorem 1) that under a settled knowledge graph the Q-table converges to the optimal value function of the underlying MDP, so that we recover the Watkins-Dayan guarantee in equilibrium. We prove (Theorem 2) that under a stationary knowledge graph with a mild gap condition, the action-selection rule produces actions satisfying the compiled aspiration on a non-negligible fraction of decisions, so that the framework is useful before convergence. We prove (Theorem 3, with a graph-distance-parameterized refinement in Theorem 3-G) that the value of the policy under a perturbed knowledge graph differs from its value under the true knowledge graph by an amount bounded in the perturbation magnitude, so that performance remains bounded under the noise that real knowledge graphs carry. Through these four theorems we have addressed the four requirements named in Section 7.1: convergence, a quantitative guarantee on decisions taken before convergence, robustness to upstream error, and structural auditability.

We have demonstrated the framework on two inventory benchmarks of differing scale, run with identical code paths and identical hyperparameters. We measured the satisficing rate of CA-RS-Q on the last fifty episodes at 88 percent on Benchmark B (two SKUs, 4,356 by 16 state-action space) and 37 percent on Benchmark A (one SKU, 66 by 5 state-action space), and we observed greater effectiveness at scale rather than less. We measured the empirical value gap under a doubled news rate at approximately 2.6 percent of the trivial worst-case bound on Benchmark B and 7.1 percent on Benchmark A, supporting the robustness predictions of Theorem 3. We also report that CA-RS-Q underperforms standard Q-learning on raw episode return in stationary regimes by

approximately thirty percent. We do not claim empirical superiority on raw return. The four properties we have established for the framework are properties we cannot guarantee from raw-return maximization alone, and in operating environments where deployment requires these four properties, the raw-return cost is a price we accept.

We have established auditability as a structural property of the algorithm rather than a post-hoc reconstruction. In Section 7.7.6 we walked through a single decision, traced each contribution to the compiled aspiration back to a specific named entity and a specific edge along which the contribution traveled, and showed that a domain expert reading the trace can verify the active conditions, the severities, the confidences, and the resulting adjustment to the cost ceiling. We cannot produce the same trace for a standard Q-learning agent. We did not add auditability to CA-RS-Q as a separate feature; the audit trail follows from the same compiler that defines $\rho(s)$.

We identify three directions for future work. First, we have used a linear instance of the compiler throughout the chapter for transparency of exposition, and we expect nonlinear compilers to satisfy the Lipschitz axiom under broad conditions; an explicit construction is left open. Second, we have proved the four theorems under the tabular assumption, and we identify the function-approximation extension of CA-RS-Q as the most substantial next direction for scaling the framework to industrial state spaces. Third, the graph-distance bound in Theorem 3-G uses an edit-distance metric that admits tighter analyses under stronger graph-structural assumptions, and we expect the bound to sharpen under such analyses. We identify each direction in Section 7.8 as a specific extension of the work we present here.

We close with three claims we believe we have earned in this chapter. First, the source of the aspiration in satisficing reinforcement learning need not be arbitrary; we can take it from the structured information the operating environment already maintains. Second, the four properties we require for high-stakes deployment (convergence, a quantitative guarantee before convergence, robustness to upstream error, and structural auditability) need not be traded against one another; we have shown they coexist in a single framework when the agent's learning channel and its aspiration channel are kept structurally separate. Third, auditability that survives the algorithm rather than being reconstructed afterwards is achievable when we design the decision structure for it from the start. The framework presented in this chapter is one realization of these three claims. We expect others to follow.

We opened this chapter by setting out to address RQ3b and to examine the hypothesis that an agent which draws its standard for acceptable performance from a structured, dynamically updated description of the world maintains satisficing performance and adapts more reliably under non-stationary conditions than an agent learning from the reward signal alone. Within the scope of the inventory benchmarks we study, the evidence we present is consistent with that hypothesis: by compiling the aspiration from the Knowledge Graph and holding the value-learning channel separate from the aspiration channel, we let the agent follow a structural change and recover when we withdraw it, while preserving convergence, a finite-time satisficing rate, and graceful degradation under graph error. We are careful about what we claim. We make no claim to an advantage in raw return; in stationary, well-specified regimes the agent is deliberately conservative, and we accept that cost. We are not aware of a directly comparable robustness guarantee, and we offer ours as a contribution rather than a settled result. On that basis we offer the new knowledge we set out to produce, and we take this to contribute M(si) and K(si) to the Execution layer of the DSIM. With respect to verification and validation, we read the Frame of Reference and the formal results as offering theoretical structural validity, the inventory benchmarks as a reasonable vehicle for empirical structural validity, and the reported behavior under structural shocks as empirical performance validity within the bounds of these problems; theoretical performance validity we defer to Chapter 9.

Everything to this point we evaluate against technical performance and governance rules. What we do not yet measure is how the people on the receiving end respond. The strategies we execute eventually reach farmers, communities, workers, and businesses, who answer not through reward functions but through cognition, emotion, memory, and social dynamics that a designer cannot capture through rational-actor models. Whether an intervention succeeds in the world depends on that response. That is the work of Chapter 8, where we assess social legitimacy before deployment.

References

[1] Hallak, A., Di Castro, D., and Mannor, S., 2015, "Contextual Markov Decision Processes," arXiv:1502.02259. https://arxiv.org/abs/1502.02259

[2] Simon, H. A., 1956, "Rational Choice and the Structure of the Environment," *Psychological Review*, 63(2), pp. 129–138. https://doi.org/10.1037/h0042769

[3] Simon, H. A., 1979, "Rational Decision Making in Business Organizations," *American Economic Review*, 69(4), pp. 493–513.

[4] Watkins, C. J. C. H., and Dayan, P., 1992, "Q-Learning," *Machine Learning*, 8(3–4), pp. 279–292. https://doi.org/10.1007/BF00992698

[5] Mazouchi, M., Yang, Y., and Modares, H., 2022, "Data-Driven Dynamic Multiobjective Optimal Control: An Aspiration-Satisfying Reinforcement Learning Approach," *IEEE Transactions on Neural Networks and Learning Systems*, 33(11), pp. 6183–6193. https://doi.org/10.1109/TNNLS.2021.3072571

[6] Khandelwal, V., Yip, H. Y., and Sheth, A., 2026, "Toward Neurosymbolic Reinforcement Learning via Editable Specifications," *Proceedings of the AAAI Symposium Series*, 8(1), pp. 603–607. https://doi.org/10.1609/aaaiss.v8i1.42594

[7] Tamatsukuri, A., and Takahashi, T., 2019, "Guaranteed Satisficing and Finite Regret: Analysis of a Cognitive Satisficing Value Function," *Biosystems*, 180, pp. 46–53. https://doi.org/10.1016/j.biosystems.2019.02.009

[8] Chehade, M., Ghosal, S. S., Chakraborty, S., Reddy, A., Manocha, D., Zhu, H., and Bedi, A. S., 2025, "Bounded Rationality for LLMs: Satisficing Alignment at Inference-Time," *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, arXiv:2505.23729. https://arxiv.org/abs/2505.23729

[9] Takahashi, T., Kohno, Y., and Uragami, D., 2016, "Cognitive Satisficing: Bounded Rationality in Reinforcement Learning" (in Japanese), *Transactions of the Japanese Society for Artificial Intelligence*, 31(6), pp. AI30-M_1-11. https://doi.org/10.1527/tjsai.AI30-M

[10] Reverdy, P., Srivastava, V., and Leonard, N. E., 2017, "Satisficing in Multi-Armed Bandit Problems," *IEEE Transactions on Automatic Control*, 62(8), pp. 3788–3803. https://doi.org/10.1109/TAC.2016.2644380

[11] Michel, T., Hajiabolhassan, H., and Ortner, R., 2023, "Regret Bounds for Satisficing in Multi-Armed Bandit Problems," *Transactions on Machine Learning Research*. https://openreview.net/forum?id=QnT41ZGNh9

[12] Hajiabolhassan, H., and Ortner, R., 2025, "Online Regret Bounds for Satisficing in Markov Decision Processes," *Mathematics of Operations Research*, 51(2), pp. 1097–1119. https://doi.org/10.1287/moor.2023.0275

[13] Liang, H., and Luo, Z., 2024, "Regret Bounds for Risk-Sensitive Reinforcement Learning With Lipschitz Dynamic Risk Measures," *International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 1774–1782.

[14] Minami, A., Kono, Y., and Takahashi, T., 2021, "Contextual Exploration Using a Linear Approximation Method Based on Satisficing," arXiv:2112.06452.

[15] Arumugam, D., Kumar, S., Gummadi, R., and Van Roy, B., 2024, "Satisficing Exploration for Deep Reinforcement Learning," *RLC Finding the Frame Workshop*, arXiv:2407.12185. https://arxiv.org/abs/2407.12185

[16] Russo, D., and Van Roy, B., 2022, "Satisficing in Time-Sensitive Bandit Learning," *Mathematics of Operations Research*, 47(4), pp. 2815–2839.

[17] Uragami, D., Sonota, N., and Takahashi, T., 2024, "Social Satisficing: Multi-Agent Reinforcement Learning With Satisficing Agents," *BioSystems*, 243, p. 105276.

[18] Yongacoglu, B., Arslan, G., and Yüksel, S., 2023, "Satisficing Paths and Independent Multiagent Reinforcement Learning in Stochastic Games," *SIAM Journal on Mathematics of Data Science*, 5(3), pp. 745–773.

[19] Doshi-Velez, F., and Konidaris, G., 2016, "Hidden Parameter Markov Decision Processes: A Semiparametric Regression Approach for Discovering Latent Task Parametrizations," *IJCAI: Proceedings of the Conference*, Vol. 2016, p. 1432.

[20] Tennenholtz, G., Merlis, N., Shani, L., Mladenov, M., and Boutilier, C., 2023, "Reinforcement Learning With History Dependent Dynamic Contexts," *International Conference on Machine Learning*, PMLR, pp. 34011–34053.

[21] Han, S., Dastani, M., and Wang, S., 2026, "Neuro-Symbolic Action Masking for Deep Reinforcement Learning," arXiv:2602.10598.

[22] Xiong, W., Hoang, T., and Wang, W. Y., 2017, "DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 564–573.

[23] Zhu, D., Li, L. E., and Elhoseiny, M., 2022, "Value Memory Graph: A Graph-Structured World Model for Offline Reinforcement Learning," arXiv:2206.04384.

[24] Ng, A. Y., Harada, D., and Russell, S., 1999, "Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping," *ICML*, Vol. 99, pp. 278–287.

[25] Skalse, J., Howe, N., Krasheninnikov, D., and Krueger, D., 2022, "Defining and Characterizing Reward Gaming," *Advances in Neural Information Processing Systems*, 35, pp. 9460–9471.

[26] Lipton, Z. C., 2018, "The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability Is Both Important and Slippery," *Queue*, 16(3), pp. 31–57.

[27] Doshi-Velez, F., and Kim, B., 2017, "Towards a Rigorous Science of Interpretable Machine Learning," arXiv:1702.08608.

[28] Glanois, C., Weng, P., Zimmer, M., Li, D., Yang, T., Hao, J., and Liu, W., 2024, "A Survey on Interpretable Reinforcement Learning," *Machine Learning*, 113(8), pp. 5847–5890.

[29] Vouros, G. A., 2022, "Explainable Deep Reinforcement Learning: State of the Art and Challenges," *ACM Computing Surveys*, 55(5), pp. 1–39.

[30] Atrey, A., Clary, K., and Jensen, D., 2019, "Exploratory Not Explanatory: Counterfactual Analysis of Saliency Maps for Deep Reinforcement Learning," arXiv:1912.05743.

[31] Robbins, H., and Monro, S., 1951, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, pp. 400–407.

[32] Bertsekas, D. P., and Tsitsiklis, J. N., 1996, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA.

[33] Bertsekas, D. P., 2025, "Neuro-Dynamic Programming," *Encyclopedia of Optimization*, Springer Nature Switzerland, Cham, pp. 1–6.

[34] Puterman, M. L., 1990, "Markov Decision Processes," *Handbooks in Operations Research and Management Science*, 2, pp. 331–434.

[35] Sutton, R. S., and Barto, A. G., 2018, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, Cambridge, MA.

[36] Even-Dar, E., and Mansour, Y., 2003, "Learning Rates for Q-Learning," *Journal of Machine Learning Research*, 5(Dec), pp. 1–25.

[37] Feng, S., Yin, M., Huang, R., Wang, Y. X., Yang, J., and Liang, Y., 2023, "Non-Stationary Reinforcement Learning Under General Function Approximation," *International Conference on Machine Learning*, PMLR, pp. 9976–10007.

[38] Rigter, M., Lacerda, B., and Hawes, N., 2023, "One Risk to Rule Them All: A Risk-Sensitive Perspective on Model-Based Offline Reinforcement Learning," *Advances in Neural Information Processing Systems*, 36, pp. 77520–77545.

CHAPTER 8

ASSESSING ADAPTIVE INTERVENTIONS: COGNITIVELY GROUNDED STAKEHOLDER SIMULATION IN E-CPSS

Background and Context

In Chapter 7, we build agents that execute strategies at the timescale of the system, governed by structured knowledge and bounded by constraints we can audit. Throughout that work, we evaluate the agents against technical performance and governance rules: whether they meet their aspirations, whether they honor their limits, whether the team learns. We never ask the question to ultimately decide whether any strategy matters once it leaves our hands, which is how the people on the receiving end will respond. In this chapter, we build the capability to ask it before we act.

Institutions of every kind must commit to decisions whose success depends as much on how people respond to them as on the merits of the decision itself. A government weighing a regulation, an agency rolling out a program, a firm revising its prices or announcing an environmental commitment all face the same situation: the decision lands on a heterogeneous set of stakeholders who interpret it through their own interests, histories, and sense of what is fair, and whose reactions determine whether it is adopted, resisted, or abandoned. The difficult question for the decision-maker is therefore not only which option is best on paper, but how the affected parties will react: who will support and who will oppose, where resistance will gather and where it will harden into collective action, and whether the decision will be accepted as legitimate. These reactions turn on perception, memory, emotion, and social influence as much as on rational calculation, and we cannot read them from the technical performance of a strategy alone.

We meet this need through RQ4, and it is where we locate the open research gap. Large language models now let us populate a simulation with stakeholders who reason in something close to human terms, but the resulting simulations remain difficult to trust as evidence: their emotional behavior is a stylistic label rather than a mechanism, their reasoning cannot be inspected, and their

departures from rational self-interest are asserted rather than measured. We frame RQ4 around this gap and state our hypothesis directly: a stakeholder simulation in which a deterministic emotion and memory substrate governs each agent's internal state, and in which a neuro-symbolic layer grounds and validates every action against a shared world, produces behavior that is measurable, auditable, and distinct from rational-actor prediction in ways we can quantify, rather than behavior we can only illustrate.

Our contribution in this chapter is a cognitive architecture for language agents in which we integrate established models from psychology and behavioral economics, a circumplex model of affect over valence and arousal and a notion of bounded rationality, with large language models. We let a deterministic emotion and memory substrate govern each agent's internal state, we call a language model only to choose actions conditioned on that state, and we ground and validate every action through a neuro-symbolic layer of a knowledge graph and a rule engine against a shared, auditable world. Where researchers working on agentic systems aim only to make agents act, we place our emphasis on making what those agents do measurable and auditable. We frame and validate the architecture on a conflict-laden carbon-policy testbed of twenty stakeholder agents across four scenarios, using a capability ladder through which we remove one mechanism at a time so that we can isolate the contribution of each. We find that structured memory is the most consequential mechanism for coherent behavior, that arousal coupling measurably improves consistency, that the symbolic layer enforces a body of constraints we would not see from a compliance score alone, and that each stakeholder class departs from rationality by a distinct, measured amount.

Through this work, we construct Layer 5 of the Decision-State Information Model, the Social Assessment layer, the means by which we test a strategy against the people it will reach before we commit to it. Through this layer we supply the social-legitimacy criterion of the transition condition T rather than a lettered component of its own, and this is the only layer through which we contribute to the model in that way. With this layer in place, we close the transition condition $T(s_i \rightarrow s_{i+1})$: we advance the system from one state to the next only when a strategy is robust, preserves the knowledge produced at the state, meets its aspiration, and is socially legitimate, the four conditions together. In building this layer, we operationalize two E-CPSS design requirements from Chapter 2: Emergent Behavior, through population-level patterns of coalition and conflict

that we do not program but that arise from the agents' interaction; and Decentralized Decision-Making, through stakeholders who deliberate and act on their own interests rather than under a central rule. We carry the same commitment here that runs through the DSIM as a whole, that a strategy is not finished when it performs well on paper but when we can show, in advance and with evidence, that the people it affects will accept it.

With this chapter, we complete the five operational layers of the Decision-State Information Model. We can read the state of an E-CPSS, structure robust decisions within it, design adaptive strategies across its transitions, augment our own judgment, execute at the speed of the system, and assess social legitimacy before deployment. What remains is to show that these six contributions hold together as a single coherent framework rather than as a set of methods that happen to share a vocabulary, and to weigh that framework against the eleven requirements we established in Chapter 2. We take up that synthesis in Chapter 9.

Preface

Governments, regulators, and companies alike must commit to decisions, from a carbon policy to a product launch, whose success depends on how a diverse set of stakeholders will react, yet they have few ways to anticipate those reactions before acting. With large language models, it has become possible to simulate such stakeholders fluently, but the resulting simulations remain difficult to trust as evidence: their emotional behavior is a stylistic label rather than a mechanism, their reasoning cannot be inspected, and their departures from rational self-interest are asserted rather than measured. In this chapter, we address the following question: *how can a generative multi-agent simulation of stakeholder behavior be made measurable, auditable, and therefore usable as evidence for a decision rather than as mere illustration?* As our contribution, we propose a cognitive architecture for language agents in which we integrate established models from psychology and behavioral economics, namely a circumplex model of affect over valence and arousal and a notion of bounded rationality, with large language models. In the architecture, we hold each agent's internal state in a deterministic emotion and memory substrate, a language model is called only to choose actions conditioned on that state, and a neuro-symbolic layer of a knowledge graph and a rule engine grounds and validates every action against a shared, auditable world. Where much current work on agentic systems aims to make agents act, we place our emphasis on making what they do measurable and auditable. We frame and validate the

architecture on a conflict-laden carbon-policy testbed of twenty stakeholder agents across four policy scenarios, using a capability ladder through which we remove one mechanism at a time so that we can isolate the contribution of each. We find that structured memory is the most consequential mechanism for coherent behavior, that arousal coupling measurably improves the consistency of agent behavior, that the symbolic layer enforces a substantial body of constraints that a compliance score would conceal, and that each stakeholder class departs from rationality by a distinct, measured amount. We further find these results to be stable when we perturb the agent population through Latin hypercube persona sampling, permute the order of action, and sweep the calibration parameters, and to transfer in direction to a stronger language model from a different family. Although we frame the study on carbon policy, the architecture is domain-general; we evaluate it on a literature-grounded synthetic testbed and do not claim calibrated prediction of real-world outcomes, and we offer the architecture, together with the measurability and auditability it affords, as a step toward stakeholder simulations that a government, a regulator, or a company could use to weigh a policy or a product decision.

Research Question, Hypothesis, and New Knowledge in this Chapter

Research Question	RQ4. What are the principles and method for modeling stakeholder behavior through cognitively grounded agent simulation to support pre-deployment social legitimacy assessment of E-CPSS interventions?
Hypothesis	Hypothesis 4. By designing stakeholder agents whose emotional influence on cognition is grounded in predictive-processing principles, specifically by treating arousal as a proxy for epistemic uncertainty that dynamically scales how strongly emotional state biases memory retrieval, and by grounding all agents in a shared knowledge graph with role constraints enforced through a rule system, we produce behavioral simulations that are measurably distinct from rational-actor models and yield diagnostic pre-deployment information about stakeholder resistance before any intervention is deployed.
New Knowledge	New Knowledge 4. A cognitively grounded stakeholder agent architecture with predictive-processing-anchored memory-emotion coupling, neurosymbolic knowledge graph grounding, and automated agent

generation for pre-deployment social legitimacy assessment of adaptive interventions in Evolving Cyber-Physical-Social Systems.

8.1 Frame of Reference

Institutions of every kind must commit to decisions whose success depends as much on how people respond to them as on the merits of the decision itself. A government weighing a new regulation, a public agency rolling out a program, and a company launching a product, revising its prices, or announcing an environmental commitment all face a common situation: the decision lands on a heterogeneous set of stakeholders including citizens, customers, employees, investors, regulators, advocacy groups, and the press who interpret it through their own interests, histories, and sense of what is fair, and whose reactions determine whether it is adopted, resisted, or abandoned. For the decision-maker, the difficult question is therefore not only which option is best on paper, but how the affected parties will react to it: who will support it and who will oppose it, where resistance will gather and where it will harden into collective action, and whether the decision will be accepted as legitimate. These reactions turn on perception, memory, emotion, and social influence as much as on rational calculation, and misjudging them is costly, because a decision that is sound on its merits can still fail when those it affects do not accept it.

What a decision-maker would value, in any of these settings, is a way to anticipate such reactions before committing to a course of action, to rehearse a decision against a realistic population of stakeholders, to compare the alternatives, and to see in advance who is likely to be persuaded, who is likely to mobilize, and where the lines of conflict and coalition would fall. In this chapter we provide a step toward that capability. We develop an architecture for simulating how a population of stakeholders, each with its own goals, memory, and disposition, would deliberate and act under a proposed decision, and we intend it not for any single domain but for multi-stakeholder decision-making in general, whether the decision-maker is a government, a regulator, a public institution, or a firm.

8.1.1 Simulating Stakeholder Deliberation for Strategy and Policy Decisions

Researchers and practitioners in both public and private settings recognize the value of rehearsing a decision in this way, and they address parts of it through several established approaches, each of

which leaves part unmet. Agent-based models let researchers study how population-level patterns emerge from the interaction of many individuals, but their agents have typically been simple rule-followers or optimizers, a reasonable simplification for some questions but a poor match for the rich, narrative, and emotional reasoning that drives real stakeholder behavior. Through surveys, focus groups, and deliberative exercises we capture genuine human responses, but they are slow and costly, and through them we cannot easily explore the many counterfactual decisions a planner might wish to compare. Through economic and market models we capture aggregate outcomes well, but they tell us about efficiency or demand rather than about how stakeholders will receive a decision and whether it will hold. Researchers using large language models can now do what they could not before. Park and coauthors show that language models can populate a simulation with agents that observe events, converse, form relationships, and act in fluent language, and that believable social behavior, such as the spread of information and the coordination of group activity, can emerge from their interaction [1]. For the first time, then, a stakeholder simulation can be populated with agents that reason in something close to human terms, and a growing body of work uses such agents to study negotiation, opinion change, and collective behavior. This is what makes the present moment the right one to ask whether we can make such simulations trustworthy enough to inform real decisions.

8.1.2 Why Current Generative Simulations Are Not Yet Decision-Grade

The promise of these simulations is also the source of their central difficulty. To inform a decision that a person must defend, we must produce through a simulation conclusions a reader can trace to mechanisms they can inspect, agents they can see acting within the rules of the world they inhabit, and departures from cold rationality open to their scrutiny rather than taken on faith. Generative simulations built on language models tend to fall short on each count. Larooij and Törnberg review the emerging literature on agent-based models built with language models and argue that validation is its central unresolved challenge, because the agents' behavior is stochastic and their internal decision-making is opaque, and because reported validations often rest on surface plausibility that is only loosely tied to the underlying mechanisms [2]. Three weaknesses recur in practice. Emotion, when it is present at all, is a fixed label or a single line in a prompt rather than a mechanism that changes how an agent thinks. The reasoning that turns an event into a decision is a black box, so a reader cannot check why an agent acted as it did or whether it respected the

constraints of its world. And the departures from rational behavior that make these agents interesting are described in words rather than measured.

These weaknesses matter because the behavior a planner most needs to anticipate is exactly the behavior they obscure. We illustrate this through carbon policy. Carattini and coauthors observe that a carbon tax is a cost-effective instrument for reducing emissions whose real-world adoption has nonetheless been limited, chiefly by public opposition, and that design choices such as revenue rebates, trial periods, and the way a measure is communicated strongly shape whether the public will accept it [3]. Studying France after the 2018 Yellow Vests movement, Douenne and Fabre find that respondents would largely reject a revenue-neutral carbon tax, that they overestimate their own losses and wrongly believe the measure is regressive and ineffective, and that those who oppose it tend to discard information that conflicts with their position, a pattern the authors attribute to motivated reasoning [4]. The reactions that decided this policy's fate: the perception of unfairness, the beliefs that resisted correction, the grievance that spread and organized, are affective, memory-laden, and socially transmitted, and they are precisely what a fixed emotional label cannot represent and an opaque decision step cannot expose. Such bounded rationality, in Simon's sense of agents who satisfice under limited information and capacity rather than optimize [5], is not a nuisance to be removed from such a model but the very thing it must capture and, if it is to serve as evidence, measure. The gap we address is therefore this: we can now populate a simulation with agents that argue and react in fluent language, but we cannot yet treat the result as evidence rather than as illustration for multi-stakeholder decision-making, because we cannot audit why the agents act, confirm that they stay within their world's rules, or quantify how and why they depart from rational behavior.

8.1.3 The Proposed Architecture and Contributions

We address this gap with a cognitive architecture for language agents in which each stakeholder is given psychologically grounded internals rather than a single prompt. A deterministic cognitive substrate maintains each agent's emotion and memory and never calls the language model, while the language model is invoked only to choose an agent's actions, conditioned on the substrate's current state, and a neuro-symbolic layer of a knowledge graph and a rule engine grounds and validates those actions against a shared, auditable world. We situate the work in the lineage of cognitive architectures for language agents, in which an agent is organized around modular

memory and a structured decision process rather than treated as a bare prompt to a model [6]. Through this architecture we make no commitment to a particular domain, and we intend it for multi-stakeholder decision-making in general; we evaluate it on the deliberation over a carbon-emissions policy, which we use as a rich, conflict-laden test problem rather than as the contribution itself.

We contribute an architecture through which researchers pursue four capabilities, which we organize as four claims.

1. Arousal as an inference-control signal (C1). We treat emotion as a signal that controls reasoning rather than as a label: an agent's arousal, which encodes its epistemic uncertainty, scales how strongly affect shapes which memories it retrieves and how readily its mood is revised, with valence setting the direction of a response and arousal its strength. We find that agents whose emotion is coupled to inference in this way behave more consistently with their internal state than agents whose emotion is held fixed.

2. Neuro-symbolic auditability (C2). We record every link in the chain from an event, to the uncertainty it raises, to the memory it recalls, to the decision it drives, to the change it makes in the world, in an append-only log, and we validate every action against symbolic rules. In this way we make a run replayable and inspectable, and we can report the number of constraints the symbolic layer actively enforces, a quantity we could not see from a simple compliance rate. It is a direct response to the concern that generative simulations are black boxes.

3. Measured bounded rationality (C3). We quantify how far each stakeholder departs from a rational reference as regret, and we enter biases such as loss aversion as parameters whose magnitudes we set and defend, so that a departure from rationality is reported as a measured quantity that differs by stakeholder class rather than asserted as a label.

4. Minimal-architecture validation (C4). Because each mechanism can be removed or retained while the world and the population are held fixed, we can identify the minimal cognitive machinery sufficient to reproduce known stakeholder behavior; we find structured memory to be the most consequential single mechanism together with the

macro-level patterns, such as polarization and the formation of coalitions, that the machinery produces.

Taken together, through these properties we move a stakeholder simulation toward something a decision-maker can rely on rather than merely watch: a run that is auditable, bounded by explicit rules, and measurable in its departures from rationality. We are deliberate about the limits of this claim. We evaluate the architecture on a literature-grounded synthetic testbed, without external ground-truth data on real stakeholder behavior, and we therefore do not claim calibrated prediction of real-world outcomes; what we demonstrate is the architecture and the auditability, measurability, and mechanistic interpretability it affords. We regard this as a necessary step toward stakeholder simulations that a government, a public institution, or a firm could use to design strategies and policies and to weigh their social legitimacy before a decision is implemented, and we take the methodological groundwork for that goal to be the work of this chapter.

We organize the remainder of the chapter as follows. In Section 8.2 we review the related work across cognitive architectures for language agents, emotion and memory in artificial agents, and neuro-symbolic and auditable multi-agent systems, and we identify the gap each leaves that motivates our design. In Section 8.3 we present the proposed architecture, its per-agent cognitive cycle, its world model and rule engine, and our experimental design and measures. In Section 8.4 we report and discuss the results on the carbon-policy testbed, isolating the contribution of each mechanism and examining the social structure the architecture produces.

8.2 Related Work

We position this work within the lineage of cognitive architectures for language agents, and we organize the related work around four threads that together motivate our design. We first review how researchers structure the reasoning of a language agent through cognitive architectures, and we ask what such structures still leave out. We then turn to how artificial agents represent emotion, to how they store and retrieve memory, and finally to how multi-stakeholder systems combine symbolic constraint with learned behavior. We read each thread against a single question: what does an agent need so that a generative social simulation can be treated as evidence rather than as illustration. At the close of each thread we state what we find missing and how we respond to it

through the architecture we propose, and we are deliberate about claiming no more than our results support.

8.2.1 Cognitive Architectures for Language Agents

Researchers working on cognitive architectures have long sought a single structure that can hold an agent's memory, perception, and decision-making together. Laird, Newell, and Rosenbloom organize cognition in Soar around search through problem spaces over a production memory, with subgoaling and learning by chunking [7], and Anderson and coauthors organize ACT-R as a set of modules, for declarative memory, goals, and perception and action, coordinated by a central production system [8]. Sumers and coauthors carry this tradition into the era of large language models with the CoALA framework, through which they describe a language agent in terms of modular memory, namely working, episodic, semantic, and procedural memory, a structured space of internal and external actions, and a generalized decision-making cycle [6]. A number of researchers instantiate parts of this picture in their agent designs. Yao and coauthors interleave reasoning traces with task actions so that the two inform each other [9]; Shinn and coauthors let an agent verbally reflect on feedback and keep that reflection in an episodic buffer to improve its next attempt [10]; and Park and coauthors maintain a memory stream of observations that agents reflect on and retrieve in order to plan, and they show that the resulting agents produce emergent social behavior such as information diffusion and coordination [1].

We build directly on this modular view of a language agent. What researchers organize through these architectures, however, is memory, action, and decision; the agent's affective state, when it is present at all, sits outside the control path and does not shape how the other modules operate. What remains missing is an integrated affective signal that ties an agent's uncertainty to how it remembers and how it decides. We address this by coupling a deterministic emotion and memory loop to a neuro-symbolic substrate, and by letting emotion act as a control signal on the agent's own reasoning rather than as a piece of its content, an integration we develop in the remaining threads.

8.2.2 Emotion and Affect in Artificial Agents

Researchers in a long line of work give emotion a precise representation. Russell describes affect with two orthogonal dimensions, a valence and an arousal, in the circumplex model [11], and

Bower explains, through his associative-network account of mood and memory, why a person in a given mood preferentially recalls experiences of congruent affective tone [12]. In artificial agents, appraisal theory has been the route by which emotion becomes a mechanism rather than a description: In their EMA model, Marsella and Gratch derive emotion from an agent's appraisal of its relationship to the environment, and the resulting emotion drives coping strategies that change the agent's own beliefs, desires, and intentions [13]. In this tradition, emotion is something that alters cognition. Researchers working on emotion in large language models have largely taken a different and lighter approach. Li and coauthors show that appending emotional phrasing to a prompt, an approach they call EmotionPrompt, can improve a model's output [14]; here emotion is a stimulus added to the input, not a state that governs how the agent reasons.

We build on the dimensional and appraisal traditions, and we depart from the prompt-level treatment of emotion in language agents. What remains missing in the language-agent setting is an affective state that controls inference rather than decorates it. We address this by treating arousal as an inference-control signal: valence sets the direction of an agent's response, while arousal sets its strength, by scaling how strongly affect shapes memory retrieval and how quickly retrieved memory revises mood. We note that this use of arousal is motivated by, and formally analogous to, the precision-weighting of prediction error under uncertainty in predictive-processing accounts of cognition [15], though we make the weaker claim of an analogy and do not derive our dynamics from a free-energy principle.

8.2.3 Memory in LLM Agents

Memory has become a central concern in language-agent design, and several distinct treatments have emerged. Park and coauthors retrieve from a memory stream to plan behavior, and demonstrate through ablation that the observation, planning, and reflection components each contribute to the believability of an agent [1]. Packer and coauthors treat memory as a systems problem, managing a hierarchy of memory tiers and paging information between the context window and external storage in the manner of an operating system's virtual memory, so as to extend an agent's effective context [16]. Shinn and coauthors keep an agent's self-reflections in an episodic buffer across attempts [10], and Bower's account of mood-congruent recall motivates making retrieval depend on affect rather than on content alone [12]. Across this work, memory is studied as capacity, as persistence, or as a contributor to believability.

We build on these episodic and retrieval mechanisms, and we ask a question they leave open. What remains missing is an isolation of memory's causal contribution to an agent's decisions, measured against a rational reference in a setting where stakeholders hold conflicting interests. We address this by removing or retaining memory while holding the world, the scenarios, and the agent population fixed, and by measuring how its presence changes both the coherence of behavior and the quality of decisions; we find memory to be the most consequential single mechanism in the architecture, which is the basis for our treatment of it as load-bearing.

8.2.4 Neuro-Symbolic and Auditable Multi-Agent Systems

A final thread concerns the combination of symbolic constraint with learned behavior, and the conditions under which a simulation can be trusted. Garcez and Lamb survey neuro-symbolic computing as the integration of neural learning with symbolic representation and reasoning, undertaken precisely to recover interpretability and accountability [17]. The need for such guarantees is sharpened in generative social simulation: Larooij and Törnberg review agent-based models built on large language models and argue that these models may exacerbate rather than ease the problem of validation, because their behavior is stochastic and their internal decision-making is opaque, and because reported validations often rest on face-validity measures only loosely tied to the underlying mechanisms [2]. Alongside this concern sits a body of theory we draw on to make behavior measurable. Simon describes, in his account of bounded rationality, agents of limited computational capacity that satisfice rather than optimize [5]; Tversky and Kahneman characterize loss aversion, the tendency for losses to weigh more heavily than comparable gains [18]; and DeGroot describes how a group can move toward consensus by repeatedly forming a weighted average of one another's opinions [19].

We build on neuro-symbolic integration and on these accounts of bounded behavior, and we take the validation concern as a design requirement. Two things remain missing. First, symbolic constraints in such systems are rarely both enforced and auditable at once. We address this by recording every link in the chain, from an event, to the uncertainty it raises, to the memory it recalls, to the decision it drives, to the change it makes in the world, in an append-only change log, and by reporting constraint integrity, the number of constraints the symbolic layer actively enforces over a run, which a simple compliance rate conceals. Second, researchers usually assert departures from rational behavior rather than measure them. We address this by quantifying each stakeholder

class's departure from a rational reference policy as regret, and by entering biases such as loss aversion as parameters whose magnitudes we set and defend, so that we report bias as a number rather than assert it as a label. We model the within-turn social influence among agents as a bounded, mood-modulated form of the DeGroot averaging dynamic, which lets us analyze when the deliberation converges and when it polarizes.

8.3 The Proposed Cognitive Architecture

We couple a psychologically grounded model of emotion and memory to the reasoning of a large language model, and we hold the pair to the rules of a shared, auditable world. Through large language models it is possible to populate a social simulation with agents that read events, argue, and react in fluent language, and a growing body of work uses them to study negotiation, opinion change, and collective behavior. In most of these simulations, however, the agents remain thin in three ways that matter for studying stakeholder decision-making. First, emotion, when it is present at all, is a fixed label or a single line in a prompt, rather than a mechanism that changes how an agent thinks. Second, the reasoning that turns an event into a decision is opaque, so that a reader cannot check why an agent acted as it did, or whether it stayed within the rules of its world. Third, the departures from rational behavior that make such agents interesting are described in words rather than measured. Together, these gaps make it difficult to treat a generative social simulation as evidence rather than as illustration.

In this section, we present the proposed architecture to address these gaps. The proposed architecture is a cognitive architecture for language agents, in which each stakeholder is given psychologically grounded internals rather than a single prompt. Each agent carries an emotion and memory loop that runs in deterministic code, coupled to a neuro-symbolic substrate of a knowledge graph and a rule engine, and it uses a language model only to choose its actions. We intend the proposed architecture for multi-stakeholder decision-making in general, since its machinery makes no commitment to a particular domain. We evaluate it on the deliberation over a carbon-emissions policy, which we use as a rich, conflict-laden test problem rather than as the contribution itself.

Three objectives are critical in design, and we return to each of them when we define our measures in Section 8.3.12. First, we want emotion to act as a control signal rather than as decoration, so that an agent’s uncertainty changes how strongly it recalls the past and how readily it revises its

mood, instead of standing as a static label. Second, we want the simulation to be auditable, so that every link from an event, to the uncertainty it raises, to the memory it recalls, to the decision it drives, to the change it makes in the world, is recorded and can be traced. Third, we want the effect of each cognitive mechanism on behavior to be a measured quantity, so that we can ask which minimal set of mechanisms is sufficient to reproduce known stakeholder behavior, together with the macro-level patterns, such as polarization and coalition formation, that follow from it.

We model each stakeholder as a language agent that is embedded in a cognitive architecture rather than as a bare prompt to a language model. In the proposed architecture, a deterministic cognitive substrate maintains the agent's affect and memory and never calls the language model, while the language model is invoked only to propose an action, conditioned on the substrate's current outputs, and a symbolic layer grounds and validates that action against a shared, auditable world. We situate the proposed architecture within the lineage of cognitive architectures for language agents. We present two figures to describe it. We present Figure 8.1 to show the whole architecture together with the per-turn cycle that each agent runs, and we present Figure 8.2 to detail the agent's internal loop, showing how affect, memory, and the neuro-symbolic decision interact, and the two points at which arousal exerts its control. We label the stages of the per-turn cycle Steps 1 to 5, and we use these labels throughout the chapter.

8.3.1 Two Reasoning Layers and the Per-Turn Cycle

In Figure 8.1, we show the architecture at three levels. At the top, we place the society of agents, in which each agent acts in the shared world and, along the dashed connections, influences the others. In the middle, we expand a single agent into the five-stage cycle it runs each turn, namely Observe, Recall, Feel, Decide and act, and Settle, and we mark each stage by whether it runs in deterministic code, calls the language model, or applies a rule check. At the bottom, we place the shared environment that every agent reads from and writes to, comprising the stream of events, the world state held as a structured map, and a history log in which every change is saved with its time and cause. We use this figure as the map for the rest of Section 8.3: we describe the per-turn cycle in Sections 8.3.1 and 8.3.2, the agent's internal state in Sections 8.3.3 to 8.3.8, and the shared world in Sections 3.9 and below.

In Figure 8.2, we open the same agent to show how affect, memory, and the neuro-symbolic decision interact within a single turn. On the left, a perceived event raises the agent's uncertainty, which sets its arousal, the control signal at the center of our design. We draw arousal as acting at two points, which we label by their gains: through α it sharpens how strongly recalled memories are weighted, and through η it governs how far recalled memories move the agent's mood, a state we represent by its valence and its arousal. On the right, the language model proposes an action conditioned on the agent's current mood and recalled memory, and the rule engine then allows, adjusts, or blocks that action before it is written back to the shared world. We return to the two control points, α and η, when we define the memory and mood updates in Sections 3.4 and 3.5.

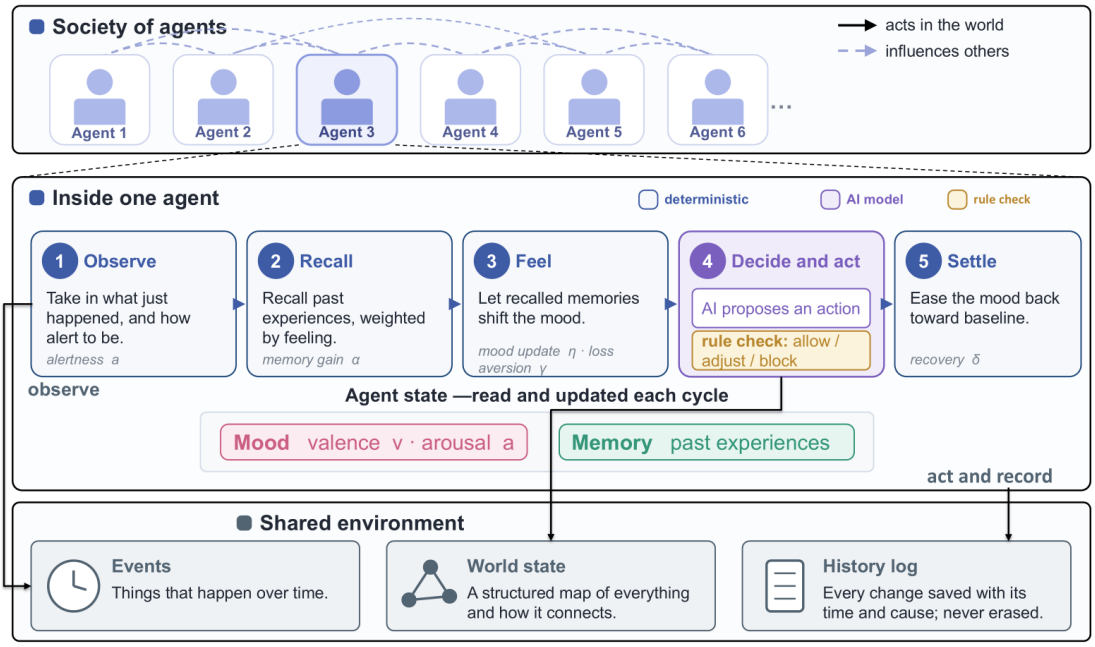


Figure 8.1: Overview of the Proposed Cognitive Architecture.

We build every agent from two stacked layers, and we separate them deliberately. The first layer is a deterministic cognitive substrate that holds and updates the agent's mood and memory, retrieves past experiences, and validates actions, all in plain code that makes no call to the language model. The second layer is the language model itself, which the agent invokes only to propose an action, and which it conditions on the substrate's current outputs expressed in natural language. We separate the two so that the parts of the agent we most want to inspect and reproduce, namely its affect, its memory, and the rules it must obey, do not depend on the stochastic output of a

language model, while the open-ended judgment that a language model is good at is confined to a single, well-defined step. The language model never sees the raw constants of the affective dynamics; it sees their interpreted result, for example that the agent is pessimistic and highly uncertain, so that it reasons over the agent's state rather than managing it.

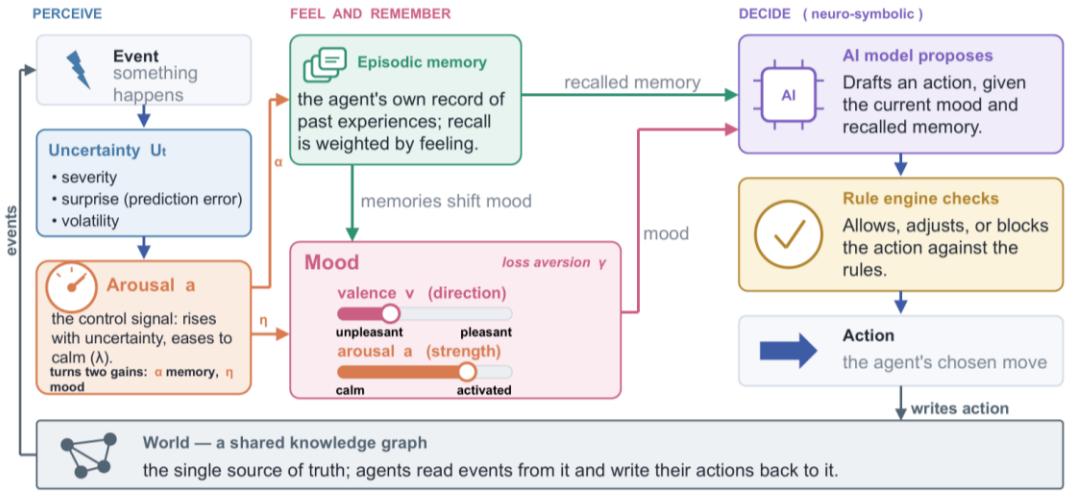


Figure 8.2: The Internal Loop of a Single Agent

Each turn, an agent runs a fixed five-stage cycle (the five stages of Figure 8.1), and the order of the stages reflects how we want affect, memory, and decision to interact. In Step 1 the agent observes the events of the turn and updates its arousal, so that the turn begins by registering how uncertain the world has become. In Step 2 it retrieves memories, weighted by its current affect, so that an aroused agent recalls a different past than a calm one. In Step 3 it lets the retrieved memories update its mood, closing the loop between what the agent remembers and how it feels. In Step 4 it proposes an action with the language model and commits the action that survives validation against the symbolic rules. In Step 5 it relaxes its mood toward a persona baseline, which we place last, after the agent has decided and recorded the turn, so that the agent reacts fully in the moment and only then begins to recover. Only Step 4 calls the language model, while Steps 1 to 3 and Step 5 are deterministic. This keeps all but one step of the cycle inspectable and free of model cost.

8.3.2 The World Model

We represent the world as a typed knowledge graph that serves as the single source of truth for a run. The graph holds five entity types, namely the stakeholders, the policies under deliberation,

the compliance records that bind a stakeholder to a policy, the simulation events that the scenario emits, and the financial states of the economic actors. We connect these entities with twelve relation types that, for example, place a stakeholder in or out of a policy's scope, attach a financial state to a stakeholder, bind a compliance record to a policy, record that an event affects a stakeholder, and record an alliance between two stakeholders. We treat the graph as a dynamic state graph with temporal provenance rather than as a reasoning engine, which is to say that the schema of entity and relation types stays fixed while the instances, their attributes, and their relationships evolve every turn.

We make this world auditable by construction. Every world-changing action writes a mutation to the graph, and we append a record of that mutation, with its old value, its new value, and its cause, to an append-only change log, so that a reader can replay a run and trace why any quantity changed. We hold an agent's action to be real only when it has produced a logged mutation, which means that language-model narration without a corresponding mutation is treated as dialogue and not as an action. Through this invariant we answer the objection that generative social simulations are black boxes, since the consequential history of a run lives in the change log, not in the model's prose. When an agent assembles its decision prompt in Step 4, it reads grounded facts from the graph, including the list of actions we permit its stakeholder class to take, so that **we never ask** the agent to act outside the world's recorded state. Concrete actions also become events that other agents observe on later turns, which we call the world channel and distinguish in Section 8.3.8 from the channel through which agents influence one another.

8.3.3 Mood and Arousal

We depart from agents that treat emotion as a fixed label or a line in a prompt, and instead give each agent an affective state that drives its later reasoning. Following a circumplex view of affect, we write the affective state at turn t as a pair of a valence and an arousal,

$$m_t = (v_t, a_t), \qquad v_t \in [-1, 1], \quad a_t \in [0, 1]. \tag{8.1}$$

Here m_t is the agent's mood, the valence v_t encodes how much better or worse the agent's situation is than it expected, from strongly negative at -1 to strongly positive at $+1$, and the arousal a_t encodes the agent's epistemic uncertainty, from calm at 0 to highly activated at 1. We treat arousal in particular as the agent's sense of how unsettled and unpredictable its world has become, because

it is this sense that we use to control the agent's downstream inference. We place this arousal signal at the center of Figure 8.2, from which its two downstream effects on memory and on mood branch.

At the start of each turn we update arousal with a first-order filter that blends the previous turn's arousal with a fresh uncertainty signal,

$$a_t = (1 - \lambda)\, a_{t-1} + \lambda\, U_t. \tag{8.2}$$

In this update a_{t-1} is the arousal carried over from the previous turn, U_t is the uncertainty raised by the current turn's events, defined just below, and $\lambda \in (0,1)$ is a per-persona update rate that sets how much weight the new signal receives. The complementary factor $(1 - \lambda)$ is therefore the carry-over of the previous arousal, and it governs how quickly arousal decays back toward calm on quiet turns, with a half-life set by λ.

The uncertainty signal U_t summarizes how unsettled the world is this turn. We compute it as a clipped, weighted sum of three quantities, with a small floor on quiet turns,

$$U_t = \mathrm{clip}\big(w_{\mathrm{sev}}\, S_t + w_{\mathrm{surp}}\, \mathrm{Surprise}_t + w_{\mathrm{vol}}\, \mathrm{Vol}_t,\ 0,\ 1\big). \tag{8.3}$$

Here $S_t \in [0,1]$ is the relevance-weighted severity of the events that affect the agent this turn; $\mathrm{Surprise}_t$ is a prediction error, the average over the turn's relevant events of the gap $|\text{event valence implication} - v_t|$ between what an event implies and what the agent currently feels, so that the same bad event surprises an optimistic agent more than a pessimistic one; $\mathrm{Vol}_t \in [0,1]$ is the recent volatility of the events the agent has seen; and the nonnegative weights w_{sev}, w_{surp}, and w_{vol} set the relative contribution of severity, surprise, and volatility. The clip holds U_t in $[0,1]$, and a small baseline floor keeps a quiet turn from driving arousal all the way to zero. We treat arousal as an inference-control signal rather than as a second mood label: the valence sets the direction of the agent's response, while the arousal sets its strength, by scaling how strongly the agent recalls affect-laden memories in Step 2 and how fast its memory revises its mood in Step 3. In Figure 8.2 we show this mechanism and the two points in the cycle at which arousal exerts its control.

8.3.4 Affect-Modulated Memory

We give each agent an episodic memory, and we make recall depend on the agent's affect, because a stakeholder under stress does not draw on the same memories as one at ease. This dependence is the first of the two arousal control points marked in Figure 8.2: through the gain α, a more aroused agent weights the affective match of a memory more heavily when it decides what to recall. Every memory record stores the content of an experience together with its valence, its arousal, a salience weight, and the turn at which it was written, alongside the recency, entity, policy-tag, and goal-congruence fields that the retrieval rule reads. In Step 2 the agent builds a query, by one fixed template that assembles the query from the current event and the agent's top-goal keywords. We construct the query deterministically, neither by hand nor by the language model, so that retrieval is reproducible and the same state always recalls the same candidate memories. The agent then scores each stored memory i by

$$\text{Score}_i = w_{\text{sem}}\,\text{Sem}(q_t, x_i) + \alpha(a_t)\,w_{\text{aff}}\,\text{Aff}(m_t, \mu_i) + w_{\text{rec}}\,\text{Rec}_i + w_{\text{sal}}\,\text{Sal}_i + w_{\text{tag}}\,\text{Tag}_i + w_{\text{goal}}\,\text{Goal}_i. \quad (8.4)$$

In this score, q_t is the query the agent has just built and x_i and μ_i are the content and the affect of memory i; $\text{Sem}(q_t, x_i)$ is the semantic match between the query and the memory content; $\text{Aff}(m_t, \mu_i)$ is the affective congruence between the agent's current mood m_t and the memory's affect, in the spirit of mood-congruent recall; and Rec_i, Sal_i, Tag_i, and Goal_i score the memory's recency, its salience, the overlap of its entities and policy tags with the query, and its congruence with the agent's goals. The nonnegative weights $w_{\bullet}$ set the standing share of each term and the always-on weights sum to one; the goal term carries weight only for personas in which we activate confirmation bias, as we describe in Section 8.3.6, and a further confidence term is defined but held inactive in the runs reported here. The arousal gain $\alpha(a_t)$ increases linearly with arousal and multiplies only the affect term, so that w_{aff} sets the standing share of affect in the score and $\alpha(a_t)$ raises that share when the agent is uncertain. This is the first place at which arousal acts as a control signal: a calm agent retrieves on semantic relevance, while an aroused agent retrieves on affect, and so recalls a more emotionally charged version of its past. After the agent commits an action, it writes one episodic memory of the outcome each turn, tagged with its current mood and a coarse appraisal of whether the action succeeded or was rejected. We give curated seed memories and

significant, world-changing events a high salience, so that an agent's formative identity and the consequential moments of a run stay retrievable and are not drowned out by routine turns. When we switch the memory off, by skipping retrieval and the mood update that follows, we obtain the no-memory variant we use in the ablation of Section 8.3.11.

8.3.5 Mood Feedback and Recovery

In Step 3 the memories that the agent has just recalled feed back into its mood, so that what an agent dwells on shapes how it feels. We write this as a gain-controlled update,

$$m_{t+1} = \big(1 - \eta(a_t)\big)\, m_t + \eta(a_t) \textstyle\sum_i w_i\ \tilde{\mu}_i, \qquad \eta(a_t) = 0.10 + 0.30\, a_t. \tag{8.5}$$

In this update m_t is the agent's current mood and m_{t+1} its mood after the feedback; w_i are the normalized retrieval weights of the recalled memories, with $\sum_i w_i = 1$, so that more strongly retrieved memories pull harder; $\tilde{\mu}_i$ is the effective mood implication of memory i, namely its valence and arousal; and $\eta(a_t)$ is the update gain, the fraction of the way the mood moves from its current value toward the recalled memories' average. The gain $\eta(a_t)$ rises with arousal over the range 0.10 to 0.40, which is the second place at which arousal sets the strength of the agent's response: an aroused agent revises its mood quickly and is emotionally fragile (which is the second of the two control points drawn in Figure 8.2), while a calm agent is resilient and moves little. Following prospect theory's account of loss aversion, we set $\tilde{\mu}_i = \gamma\, \mu_i$ for negative-valence memories, with γ acting on the valence channel only, so that adverse memories move valence more than favorable memories of equal size; γ is one for a neutral persona and about two for a loss-averse one. This update revises both the valence and the arousal channels. Taken together with the world-driven update of Equation (8.2) and the recovery step below, this means that arousal is touched three times in a turn, once by the world, once by retrieved memory, and once by recovery. We analyze the stability of this loop separately, and we find that the emotional spirals it can produce are transient rather than divergent, because we bound the update gain, clamp the state, and pull the agent back through the recovery step. Once per turn, and only once, the agent also receives a single bounded social influence on its mood, which we define in Section 8.3.8.

In Step 5, as the true last step of the cycle, the agent's mood relaxes toward its persona baseline,

$$m_t \leftarrow m_t + \delta\, (m_{\text{baseline}} - m_t). \tag{8.6}$$

Here m_{baseline} is the persona's resting mood and $\delta \in (0,1)$ is a small per-persona recovery rate, so that the affect an agent accumulates within a turn decays a fraction δ of the way back toward baseline each turn rather than drifting without bound. We list the fixed constants λ, γ, δ, and the coefficients of $\alpha(a_t)$, together with the retrieval weights, in Table 8A1 in Appendix 8A. When we hold the affective state constant, by fixing the mood and disabling Equations (8.2), (8.5), and (8.6), we obtain the constant-affect variant we use as the direct test of whether arousal, acting as a control signal, changes behavior.

8.3.6 Cognitive Biases as Four Addressable Layers

We model cognitive biases as mechanisms that we can switch on one at a time, rather than as adjectives we attach to a persona, because we want the strength of a bias to be a number we set and defend rather than a behavior we hope the language model will display. We place each modeled bias at one of four layers of the agent. At the parameter layer we alter the affective dynamics, for example through the loss-aversion multiplier γ, or a low arousal rate λ for a status-quo persona that is slow to become alarmed. t the retrieval layer we alter the scorer weights of Equation (8.4), for example by raising the recency weight to produce recency or availability bias, or the goal-congruence weight to produce confirmation bias. The rule layer adds symbolic gates, for example an anchoring cap that limits how far a revision may move from a precedent, or an ambiguity penalty, which we describe in Section 8.3.7. At the prompt layer we add an interpreted bias fragment to the agent's decision prompt, for example optimism or moral absolutism over sacred values. We derive each bias as an explicit chain, from a labeled bias, to the mechanism that implements it, to the parameter that sets its magnitude, to the justification for that magnitude, and we let each layer be enabled or disabled on its own. More broadly, we hold every constant in the architecture to one of three standards: we anchor it to a result in the behavioral-science literature, it expresses an ordinal or structural claim whose magnitude we calibrate, or we sweep it in the sensitivity analysis of Section 8.3.11. We summarize the biases, their layers, their parameters, and their justifications in Table 8A2 in Appendix 8A.

8.3.7 The Neuro-Symbolic Rule Engine

We constrain the language model's freedom with a symbolic rule engine, both so that an agent cannot take an action its role forbids and so that every binding of an action to the rules is recorded.

In Step 4 the agent first proposes an action with the language model, grounded in the permitted-action list and the facts it reads from the knowledge graph, and the rule engine then validates that proposal through four ordered stages. This is the decision stage shown on the right of Figure 8.2, where the rule engine allows, adjusts, or blocks the proposed action before it is written to the world. The hard stage checks that the agent's stakeholder class may take the action at all. The policy stage checks policy-specific conditions, namely eligibility, the grant cap, the penalty gate, whether the policy is active, and whether the target is in scope. The soft stage bounds the magnitude of an otherwise permitted action. The domain stage evaluates outcomes such as the union strike. A hard failure vetoes the action: the agent is re-prompted once with the reason, and if it still proposes an invalid action, it falls back to a safe default. A soft or policy failure clips the action. For instance, the grant-cap rule clips an approved grant down to the policy's cap amount, comparing the requested and cap values after coercing them to a common numeric form, so that a mismatch between a string and an integer cannot let an over-large grant slip through. We write every veto and clip to the change log.

One domain rule carries much of the collective-action behavior, and we describe it here because the results turn on it. Each turn, a union accumulates grievance in proportion to the tension a policy creates between its wage and climate goals, so that a job-threatening policy fills a wage-prioritizing union's grievance quickly while a climate-leaning union accumulates little. A strike fires when the union's accumulated grievance, combined with a weighted contribution from its negative mood, crosses a mobilization threshold θ. We set θ ordinally by union type, as we describe in Section 8.3.9, and the strike itself is a world-changing action that mutates union and compliance state and is written to the change log like any other.

We are careful about how we measure the value of this layer. Because a permitted, well-formed action satisfies the rules trivially, the rate of rule compliance on the committed actions is near one whether the rules are present or absent, and so it does not separate the two cases. We therefore measure constraint integrity instead, which we define as the number of constraints the symbolic layer actively enforces over a run. With the policy, soft, and domain rules present this count is substantial, while when we disable those rules it falls to zero even though compliance still reads one; so through constraint integrity we record exactly the work the layer performs and that we could not see from the compliance rate. We give the measure in Section 8.3.12. Disabling the

policy, soft, and domain rules, while keeping the hard role-permission gate, gives the symbolic-off variant of Section 8.3.11.

8.3.8 Orchestration and Social Influence

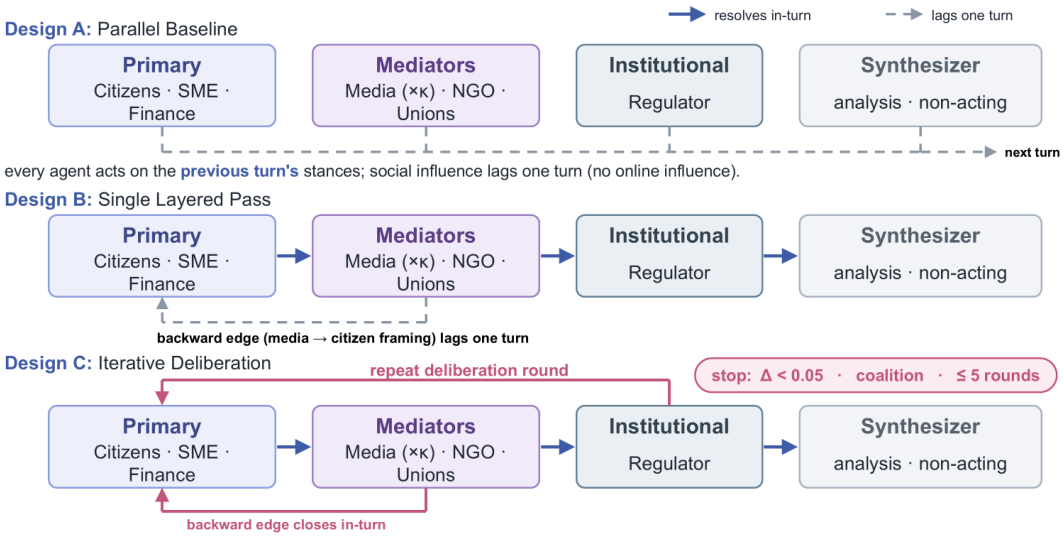


Figure 8.3: The Three Orchestration Designs.

We order the agents into three acting layers, namely the primary stakeholders, the mediators, and the institutional actors, and we add a synthesizer as a non-stakeholder analysis layer that summarizes positions without itself acting. Under our headline protocol, which we call Design C, we run the deterministic substrate once per turn and then repeat a deliberation round, in which agents blend their stances under social influence and re-decide. We adopt this iterative protocol because it lets the backward influences, such as a media frame reaching citizens after the citizens have already spoken, close within the same turn across rounds, which is what allows polarization and coalition cascades to emerge rather than being imposed. The turn ends when the per-round change in stances falls below a tolerance $\varepsilon = 0.05$, when coalition support passes a threshold, or after at most five rounds. Across rounds only the expressed stances blend, in the manner of a DeGroot opinion dynamic, while the mood takes its single bounded nudge during the once-per-turn substrate pass. We retain two further protocols only as comparison points: Design A, in which agents act in parallel and influence lags by one turn, and Design B, a single layered pass without re-deliberation. We present Figure 8.3 to contrast the three protocols. They differ in when social

influence takes effect and in whether the layered pass repeats within a turn: Design A applies influence with a one-turn lag, Design B resolves a single pass with its backward edges lagging, and Design C, our headline, repeats the pass within a turn until the stances settle.

We keep two cross-agent channels distinct. The world channel of Section 8.3.2 carries what actually happened, since concrete actions become events that others observe. The influence channel carries what others think and say, namely stance, framing, and sentiment. We set the influence that agent i exerts on agent j at turn t to

$$w(i \to j, t) = \text{base}(i \to j) \cdot \text{trust}(j \to i) \cdot g_j(t). \tag{8.7}$$

In this weight, $\text{base}(i \to j)$ is the structural capacity of the edge from i to j, which we multiply by a factor κ on the edges out of media actors so that media voices carry further; $\text{trust}(j \to i)$ is homophily, derived from the goal and ideology congruence between the two agents, so that agents listen more to those they agree with; and $g_j(t)$ is the susceptibility of the recipient j, taken in our main runs as a standalone per-persona parameter. We normalize the incoming weights so that each agent retains a dominant self-weight, and we inject the highest-weighted sources' stances into the recipient's decision prompt, applying the single per-turn mood nudge bounded at no more than about a fifth of a turn's mood movement, so that an agent's own dynamics dominate and its affect does not simply collapse onto its neighbors. We record an alliance between two agents when their accumulated alignment crosses a fixed threshold, which we set at 0.50, and we read stable, polarized alliances as the emergent social structure of a run.

8.3.9 Stakeholder Personas

We instantiate seven stakeholder classes, namely citizens, small and medium enterprises, financial institutions, media, unions, non-governmental organizations, and regulators, as twenty agents, with the synthesizer of Section 8.3.8 sitting outside this roster as an analysis layer. We hand-craft every agent and fix it before a run, and we use the same twenty agents across all scenarios and conditions, so that any difference we observe between runs comes from the mechanisms we vary and not from the cast. A persona sets the agent's stakeholder class and acting layer, its goals and the weights over them, its risk tolerance, its baseline mood, its affective constants λ, δ, and γ, its bias configuration across the four layers of Section 8.3.6, including its retrieval-weight overrides and its susceptibility, and its class-specific parameters, such as the media amplification κ, the union

mobilization threshold θ, and the lending capacity of a financial actor. We set the union threshold θ ordinally following the logic of the labor-relations literature on militancy [20, 21]: an advocacy or social-movement union, for which mobilization is a primary tool, takes a low threshold; a loss-averse industrial union takes a middle one; and a skilled-craft union that strikes only as a last resort takes a high one. We also give each agent a small set of affect-tagged seed memories that follow the record schema of Section 8.3.4, and that we check for affect consistency, so that the agent begins a run with a plausible history rather than a blank one. We summarize the roster and the persona schema in Table 8A3 in Appendix 8A.

8.3.10 The Carbon-Policy Deliberation as a Test Problem

Using the deliberation over a carbon-emissions policy as a test problem, we exercise the proposed architecture in a setting where stakeholders hold conflicting interests over a small set of consequential actions on a shared world. We run four scenarios that differ in the policy instrument under deliberation. In BASE-0 there is no new instrument and only an information campaign, which gives a counterfactual against which we read the other three. In TAX-1 a flat carbon tax is levied on the small and medium enterprises, set at fifty-five dollars per tonne of carbon dioxide equivalent with a first-year rebate. In INC-2 a green-capital incentive offers a capital-expenditure grant and a concessional loan. In HYB-3 the tax and the incentive run jointly, as a stress test of the architecture under two simultaneous pressures. Each run lasts forty-five turns, where one turn stands for about two weeks, and an event scheduler injects each scenario's timeline of announcements and shocks at fixed turns, for example a fuel-price rise, a supply-chain compliance letter, a finance liquidity squeeze, and a viral media story. Because we hold the scenario, the horizon, and the schedule fixed across conditions, we can attribute the differences between runs to the agents' mechanisms rather than to the world they face. We choose this test problem because it sets literature-grounded coalitions against one another, it supplies scheduled shocks that should drive the affective dynamics, and it confines the agents to a small, auditable repertoire of actions on a shared world. We summarize the scenarios, their instruments, and their shocks in Table 8A4 in Appendix 8A.

8.3.11 Experimental Design and Reproducibility

We evaluate the architecture with a capability ladder. We run the complete architecture together with six reduced variants, and in each we remove or replace exactly one mechanism: in the no-memory variant we remove memory; in the constant-affect variant we freeze the affective gains, and we use it as our direct test of arousal as a control signal; in the symbolic-off variant we disable the policy, soft, and domain rules, keeping only the hard role-permission gate; in the arousal-coupled variant we change the susceptibility pathway in the influence channel; and in two scripted baselines — a rational agent that selects actions by utility and a reactive agent that responds only to the latest event — we replace the language-model decision in order to bracket the ladder. Because every variant operates on the identical world, scenarios, and twenty agents, we can attribute any change in behavior directly to the single mechanism that was changed. We run this ladder across the four scenarios of Section 8.3.10, over a grid of forty experimental cells and one hundred thirty-six runs, with three seeds for the exploratory comparisons and five for the headline comparisons.

Each agent's decision is produced by a Qwen3-class model (qwen35-fp8), which we serve locally and which drives all twenty agents, and we call it at temperature zero and run the campaign serially, so that a run is determined by its seed. We report behavior as a mean with a confidence interval over seeds, and we are explicit about why. The symbolic causal chain reproduces exactly, in that a fixed seed replays the change log's entity, field, and cause structure identically, while the neural substrate of language-model text and vector retrieval is not numerically identical even at temperature zero. We regard this split as a feature rather than a caveat, because it is a concrete demonstration of the auditability we set out to achieve: the mechanism layer is exactly reproducible even when the connectionist substrate beneath it is not. To judge the agents' outputs we use an independent model from a different family, GPT-4.1-mini, so that the judge does not share the agents' own preferences. Finally, we check that our findings are not artifacts of incidental choices in three ways: we replace the citizen and small-enterprise agents with alternative persona pairings drawn by Latin hypercube sampling, we permute the order in which agents act within each layer, and we sweep internal calibration parameters across their plausible ranges. We also re-run a slice of the design on a stronger model from a different family, OpenAI's GPT-4.1, under a

decoding format matched to the primary model, so that the only difference between the two is the capability of the underlying model.

We run the full study on a single high-memory GPU, an NVIDIA H100 with 80 GB of memory, and we served the primary agent model with a high-throughput inference engine (vLLM) on that device, while we access the independent judge model and the frontier comparison model through a commercial application interface. We used approximately 1,350 agent calls when we simulated one deliberative cell with twenty agents over forty-five turns and counted deliberation within each turn. The independent judge made 75,153 scoring calls across the principal campaign. We used approximately 26,000 additional calls for the robustness study and 13,000 additional calls for the cross-model study. Because we execute every call serially at temperature zero to preserve determinism, a complete robustness campaign of this kind occupies many hours of wall-clock time. We report the implications of this computational footprint in Section 8.5.2.

8.3.12 Evaluation Measures

We report a small set of measures, each tied to one of the objectives discussed in Section 8.3, and we log every one of them directly from the per-turn traces and the change log. To show that emotion is doing work, we measure emotion-behavior coherence as the rank correlation between an agent's expressed dialogue sentiment, scored by the independent judge, and its internal valence, and we read this correlation as a within-condition quantity across the agents of a condition rather than against an absolute target; the two scripted baselines reach a coherence of one by construction, because their dialogue is templated to match their stance, so we treat them as reference points and exclude them from coherence comparisons. To show that bias is a measured quantity, we measure bounded rationality as regret against a rational reference policy that places a soft-max over the agent's permitted actions,

$$\pi_{\text{rat}}(a) \propto \exp(U(a)/\tau), \tag{8.8}$$

$$R = U(a^*) - U(a_{\text{chosen}}). \tag{8.9}$$

Here $\pi_{\text{rat}}(a)$ is the probability the rational reference policy assigns to action a; $U(a)$ is the agent's goal-weighted utility of action a, which is distinct from the uncertainty signal U_t of Equation (3); τ is a temperature that sets how sharply the reference policy favors the best action; a^* is the utility-maximizing permitted action; a_{chosen} is the action the agent actually committed; and the regret R

is the utility the agent left on the table by choosing a_{chosen} instead of a^*. We take regret as the headline measure, because it is defined on a single decision, and we report the divergence between the agent's action distribution and π_{rat} as a secondary measure where aggregation yields a distribution. To show that the symbolic layer earns its place, we measure constraint integrity, as defined in Section 8.3.7, as the number of constraints actively enforced per run, which falls to zero when the policy, soft, and domain rules are disabled even as compliance stays at one. We measure memory utilization as the precision and recall of memory retrieval against relevance, read from the retrieval traces with relevance scored by the independent judge. We read the coalition structure of a run from the alignment threshold of Section 8.3.8, and we compare the coalitions that form against literature-derived priors as a check on plausibility. We report these measures across the capability ladder and the four scenarios.

8.4 Results and Discussion

In this section, we present the results obtained by applying the proposed emotion, memory, and symbolic architecture to a multi-agent carbon-policy simulation. The simulation comprises twenty agents drawn from seven stakeholder classes (citizens, small and medium enterprises, financial institutions, media, non-governmental organizations, unions, and regulators), who deliberate and act over a forty-five-turn horizon across four policy scenarios: a baseline (BASE-0), a carbon-tax scenario (TAX-1), an incentive scenario (INC-2), and a hybrid scenario (HYB-3). As our central evaluation instrument, we use a *capability ladder*. We run the complete architecture alongside six reduced variants, each of which removes or replaces exactly one mechanism. Because every variant operates on the identical world, scenarios, and agent population, we can attribute any change in observed behavior directly to the mechanism that was removed. Through this design, we move beyond demonstrating that the architecture works as a whole, and instead isolate the contribution of each component. The complete grid comprises 136 runs over 40 experimental cells and completed with zero failures. Unless we state otherwise, all the values we report are means over the four scenarios under the deliberative orchestration condition (Design C).

We adopt the iterative deliberative orchestration (Design C) as our headline configuration for a specific reason. Among the three orchestrations we implemented, it is the only one in which backward influence, such as media framing reaching citizens, closes within a turn rather than lagging. In practice, this within-turn deliberation reduces the final spread of agent positions by

roughly 16 percent relative to the parallel design, and it is the source of the emergent factional dynamics we analyze below, which the simpler designs cannot produce. Within each turn, this deliberation settles quickly: agents reach a stable set of positions in 1.49 rounds on average, and the loop never reaches its safety ceiling of five rounds, with most turns converging in a single round once positions stabilize.

Two of our central measures depend on an independent judge model, and we describe it here. To compute emotion, behavior coherence, we use a separate sentiment judge that reads each agent's generated dialogue and scores its sentiment; we then correlate that independent score against the agent's internal valence. Coherence is therefore an externally judged quantity rather than a self-report. Over the headline campaign, this judge made 75,153 scoring calls with a malformed-output rate of 0.25 percent and no transport failures, which we take as evidence that the coherence measure rests on a stable and reliable scoring process. We use the same independent-judge approach to assess the relevance of retrieved memories.

We organize the results around four contributions. We first establish that structured memory is the most consequential mechanism in the architecture (C4), and then show that coupling emotion to inference through arousal improves behavioral coherence (C1). We next quantify how far each stakeholder class departs from rational behavior (C3), and demonstrate that the symbolic layer enforces a substantial, auditable body of constraints that are otherwise invisible to a naive compliance measure (C2). We then examine the emergent social structure that the architecture produces, and close by establishing the stability of our findings under perturbation and their generalization to a stronger language model from a different model family.

8.4.1 Memory Is the Most Important Mechanism for Coherent Behavior (C4)

In Table 8.1, we report the capability ladder. For each arm, we record four behavioral measures. *Emotion, behavior coherence* is the rank correlation between an agent's expressed sentiment and its internal valence; it captures whether an agent's outward behavior is consistent with its internal state. *Mean regret* measures decision quality against a rational baseline, which we define precisely below. *Final stance variance* is the spread of agent positions at the end of the run, a measure of how polarized or converged the population is. *Convergence rounds* is the number of within-turn

deliberation rounds required to reach a stable set of positions. Our purpose in this table is to determine which mechanism, when removed, most degrades the architecture's behavior.

Before interpreting the regret column, we define the rational baseline against which it is measured. Following the formulation in Equations (8) and (9), we define a rational policy as a utility-softmax over each agent's permitted actions, written as $\pi_{rat}(a)$ proportional to $\exp(U(a)/\tau)$, where U is the goal-weighted utility of an action and τ is a temperature. For a single decision, we then measure *regret* as $R = U(a^*) - U(a_{chosen})$, the utility gap between the optimal permitted action a* and the action the agent actually took. A regret of zero denotes a perfectly rational choice, and larger values denote larger departures from utility-optimal behavior. This is the primary measure underlying contribution C3 and the regret column throughout the results.

Table 8.1: Capability ladder under Design C, reported as means over the four scenarios (source: full 136-run grid).

Arm (mechanism removed)	Coherence	Mean regret	Stance var.	Conv. rounds
full (none removed)	0.539	0.165	0.393	1.49
no_memory	0.316	0.222	0.393	n/a
constant_affect	0.434	0.174	0.416	n/a
symbolic_off	0.438	0.184	0.414	n/a
arousal_coupled	0.484	0.173	0.376	n/a
rational (scripted)*	1.000	0.080	0.139	n/a
reactive (scripted)*	1.000	0.203	0.152	n/a

*The two scripted baselines, rational and reactive, attain a coherence of 1.000 by construction because their dialogue is templated to match their stance exactly; we

Table 8.1: Capability ladder under Design C, reported as means over the four scenarios (source: full 136-run grid).

Arm (mechanism removed)	Coherence	Mean regret	Stance var.	Conv. rounds
therefore treat their coherence as a reference point rather than a measured outcome and exclude it from coherence comparisons				

We draw several observations from Table 8.1. First, removing memory produces the largest behavioral degradation of any single ablation. When we ablate the memory module (the no memory arm), coherence falls from 0.539 to 0.316, a reduction of 0.223, and mean regret rises from 0.165 to 0.222. Agents without memory thus behave both less coherently and less effectively, making worse decisions relative to the rational baseline. We identify this as the largest single-mechanism effect in the ladder, which is the empirical basis for contribution C4.

Second, we observe that the remaining mechanisms each contribute a smaller but consistent amount of coherence. Removing adaptive emotion (the constant affect arm) lowers coherence to 0.434, and disabling the symbolic layer (the symbolic off arm) lowers it to 0.438; both are intermediate between the full architecture and the memory-ablated arm. No single one of these removals approaches the magnitude of the memory effect, from which we confirm that memory, rather than emotion or symbolic reasoning alone, is the principal determinant of coherent behavior.

Third, we note how the two scripted baselines bracket the architecture on regret. The rational baseline, which by construction always selects the utility-optimal action, attains the lowest regret (0.080); the reactive baseline, which responds without deliberation, attains the highest (0.203); and the full architecture lies between them (0.165). We return to this bracketing in Section 8.4.3, where we use it to establish bounded rationality. Finally, we observe that stance variance is similar across the non-scripted arms (0.376 to 0.416) but collapses for the scripted baselines (0.139 and 0.152), because templated agents produce uniform positions; we analyze the stance-variance behavior of the deliberative design in Section 8.4.5.

To make the structural consequence of the full architecture concrete, in Figure 8.4 we present the final agent influence network for a representative hybrid-scenario run, with each agent positioned by its alignment to others and colored by its final position. Nodes are the twenty agents, positioned by mutual alignment; edges connect agent pairs whose alignment exceeds the 0.50 threshold. Green nodes form the pro-policy bloc, red nodes the hardened opposition, and the yellow node is the contested (swing) agent. We observe that through the architecture the twenty agents organize into a coherent structure: a large, internally consistent support bloc and a small, clearly separated opposition cluster. This is the kind of organized outcome the memory mechanism supports and that degrades when we remove memory.

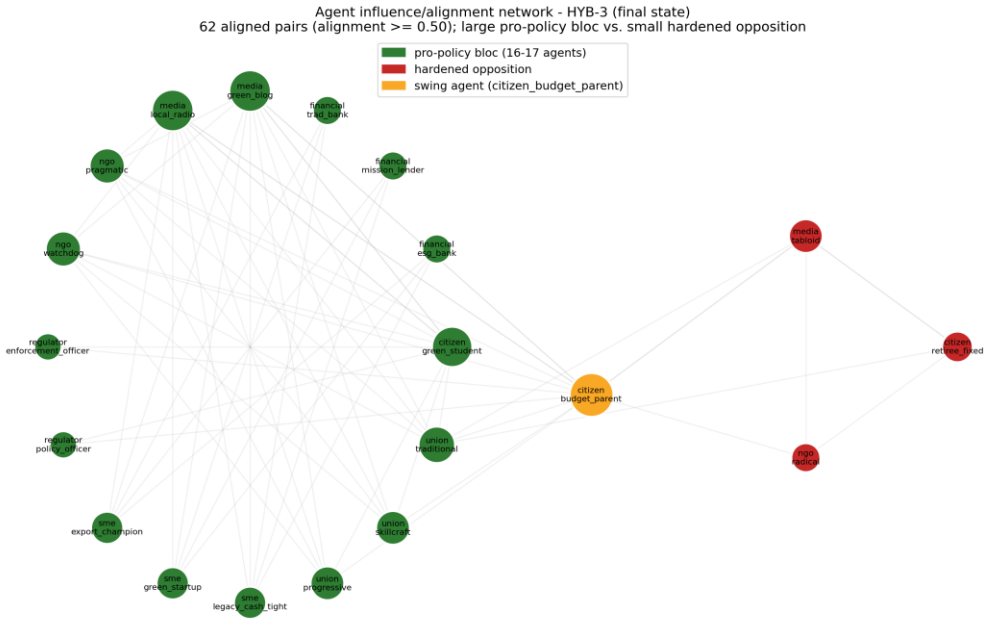


Figure 8.4: Final Agent Influence and Alignment Network for a Representative Hybrid-Scenario Run

In Figure 8.5, we complement this static view with the full temporal record, plotting the stance of every one of the twenty agents across all forty-five turns. Each line in Figure 8.5 is one agent, colored by stakeholder class; the highlighted lines mark the hardened opposition and the swing agent we analyze in Section 8.4.5. We observe two structural features that persist for the entire run:

a stable band of supporting agents in the upper region of the plot, and a small set of agents that remain in firm opposition throughout. We analyze this structure in detail in Section 8.4.5; here we use it to show that the full architecture produces durable, differentiated positions rather than uniform drift.

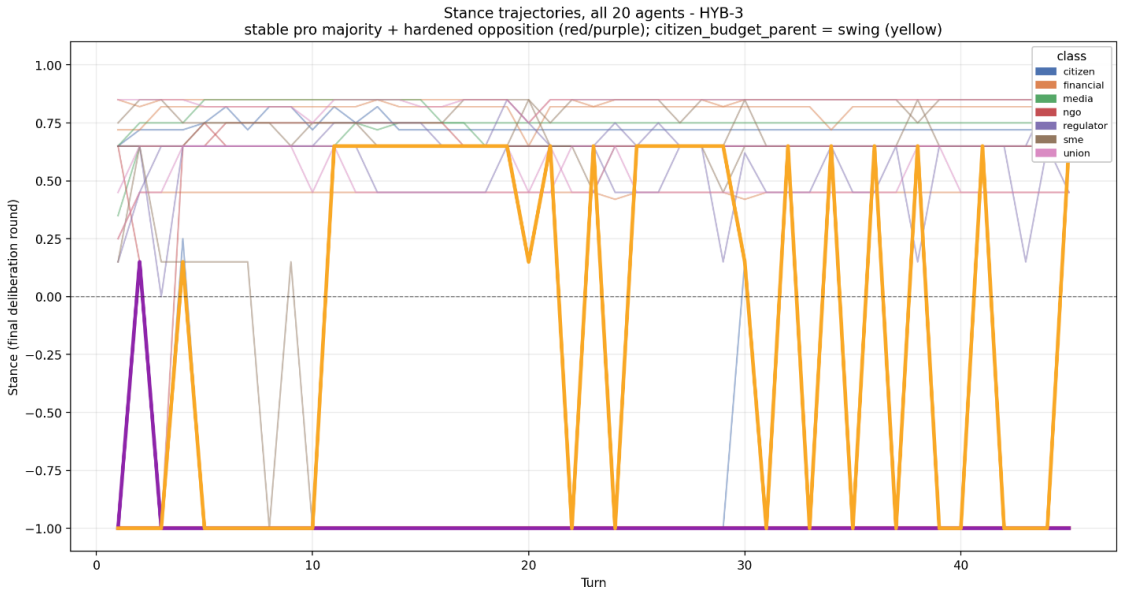


Figure 8.5: Stance Trajectories of All Twenty Agents Across the Forty-Five-Turn Horizon Under the Full Architecture (Hybrid Scenario)

8.4.2 Arousal Makes Agents Behave More Consistently (C1)

Having established the role of memory, we next isolate the contribution of arousal as an inference-control signal. In the full architecture, an agent's arousal modulates how strongly it weights newly retrieved memories and how readily it updates its mood. To test this mechanism, we compare three points on the ladder that progressively weaken the coupling: the full architecture, an *arousal_coupled* variant with reduced coupling, and a *constant_affect* variant in which emotion is held fixed. Returning to the coherence column of Table 8.1, we observe a monotonic gradient: coherence falls from 0.539 in the full architecture, to 0.484 when arousal coupling is weakened, to 0.434 when emotion is fixed entirely. The ordering is consistent and in the predicted direction. We interpret this as evidence that treating emotion as an adaptive signal which responds to

circumstance, rather than as a fixed trait, yields agents whose outward behavior is more consistent with their internal state. This gradient is the evidence for contribution C1.

We observe that this effect operates within the deliberative process itself. In Figure 8.6, we show, for four representative turns, how the twenty agents' stances evolve across the successive rounds of within-turn deliberation. In Figure 8.6, each line is one agent's stance across successive deliberation rounds; agents begin dispersed and blend toward agreement, illustrating the deliberative mechanism that arousal modulates. Agents begin each turn with dispersed positions and blend toward agreement over a small number of rounds. We find that the arousal mechanism governs each agent's receptiveness to this blending; weakening it flattens that responsiveness and, as we report in Table 8.1, reduces the coherence of the resulting behavior.

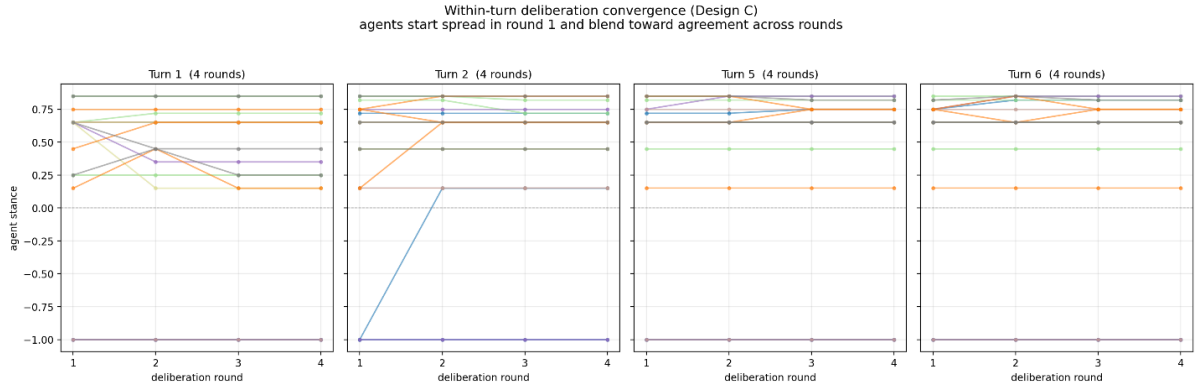


Figure 8.6: Within-Turn Deliberation Convergence Under Design C for Four Representative Turns

8.4.3 Each Stakeholder Class Departs from Rational Behavior by a Measured Amount (C3)

A central claim we make for our architecture is that through it we produce *bounded* rationality, that is, behavior that is neither perfectly optimal nor purely reactive, and that we can measure the degree of this bound. We support this claim in two steps. First, as noted in Section 8.4.1, the full architecture is bracketed by its two scripted reference points on the regret measure: the rational baseline incurs the least regret (0.080), the reactive baseline the most (0.203), and the full architecture falls between them (0.165). We read this ordering, rational below full below reactive,

as the expected signature of bounded rationality: the architecture neither matches the utility optimum nor abandons deliberation.

Second, and more importantly, we find that the departure from rationality is not uniform across agents but is specific to each stakeholder class and can be quantified. In Table 8.2, we report mean regret by class. Our purpose here is to show that through the architecture we give each class a distinct and interpretable degree of bias rather than treat all agents identically. We observe that unions depart from rational behavior most strongly (regret 0.421), which is consistent with a mechanism in which accumulated grievance drives costly collective action such as strikes; small and medium enterprises and NGOs follow (0.224 and 0.215); regulators and citizens occupy an intermediate range (0.163 and 0.094); and financial depart least (0.040). The media class attains zero regret because its agents have a dominant, role-consistent action, the publishing of framing, so their choices do not depart from the utility-optimal action; this marks the low end of the bias spectrum, in contrast to the unions at the high end. In our framework, therefore, we report bias as a measured quantity rather than asserting it as a label, which is the substance of contribution C3.

Table 8.2: Mean regret by stakeholder class for the full architecture under Design C. Regret, defined as U(a*) minus U(a_chosen), measures the utility gap between an agent's chosen action and its best permitted action; a larger value indicates a greater, class-specific departure from rational behavior.

Stakeholder class	Mean regret vs. rational baseline
union	0.421
sme	0.224
ngo	0.215
regulator	0.163
citizen	0.094
financial	0.040

Table 8.2: Mean regret by stakeholder class for the full architecture under Design C. Regret, defined as U(a*) minus U(a_chosen), measures the utility gap between an agent's chosen action and its best permitted action; a larger value indicates a greater, class-specific departure from rational behavior.

Stakeholder class	Mean regret vs. rational baseline
media	0.000

In Figure 8.7, we present the agents' action repertoire to confirm that these differentiated regret values arise from genuinely different behavior. Each colored band in Figure 8.7 is the number of agents selecting a given action on that turn; the vertical markers indicate the policy events and the two union strikes. Across the run, the twenty agents select from sixteen distinct actions, from publishing framing and lobbying to protesting, mobilizing, and, in rare cases, striking, and we observe that the mix of these actions shifts as the scenario unfolds. From this we confirm that the class-specific regret values in Table 8.2 reflect distinct behavioral patterns across classes rather than a single dominant action.

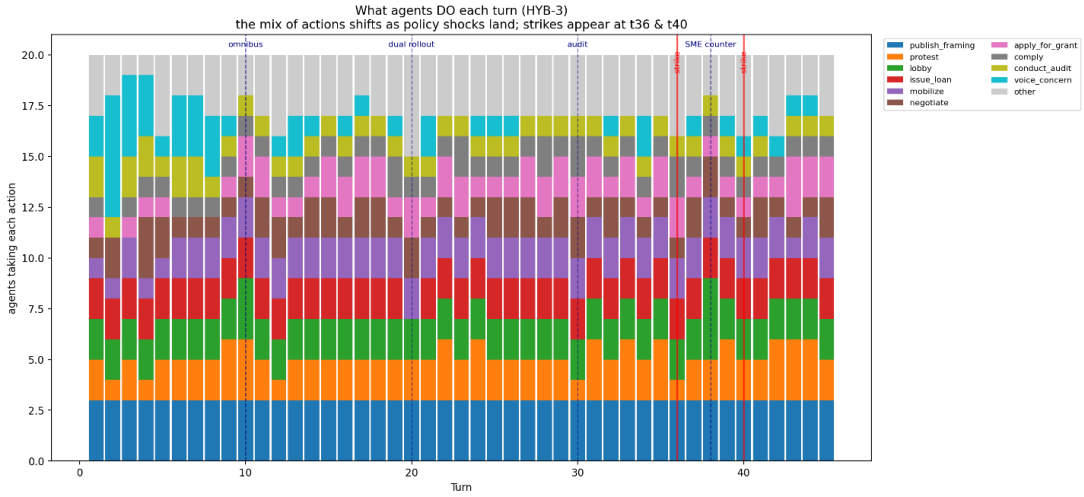


Figure 8.7: Distribution of agent actions across turns under the full architecture (hybrid scenario)

8.4.4 The Symbolic Layer Enforces Real, Auditable Constraints (C2)

We now evaluate the neuro-symbolic layer, and we begin by addressing a measurement pitfall directly. If we measure only *rule compliance*, the fraction of agent actions that satisfy the active rules, we obtain a value of 1.000 in every arm of the ladder, including the *symbolic_off* arm. The reason is that an agent operating with no rules cannot violate any, so a perfect compliance score reflects the absence of constraints just as readily as their satisfaction. We therefore conclude that compliance alone cannot demonstrate that the symbolic layer performs any work.

As the appropriate measure, we introduce *constraint integrity*, the number of constraints the symbolic layer actively enforces during a run. In Table 8.3, we report this quantity with the symbolic layer enabled and disabled. We observe that, with the layer active, between 18 and 66 constraints fire and are enforced per run depending on the scenario, whereas with the layer removed, exactly zero constraints remain in force, even though, as we see in the final column, compliance reads 1.000 in both cases. Our point is that these constraints are load-bearing: their removal genuinely changes what the agents are permitted to do, a change that the compliance score cannot detect but constraint integrity makes explicit.

Table 8.3: Constraint integrity: the number of rule firings actively enforced per run, with the symbolic layer enabled versus disabled (source: full grid). Because rule compliance is 1.000 in both conditions, it does not distinguish them; constraint integrity, not compliance, is the measure that reveals the work the symbolic layer performs.

Scenario	Constraints enforced (+symbolic)	Constraints enforced (no symbolic)	Compliance
BASE-0	18	0	1.000
HYB-3	66	0	1.000
INC-2	63	0	1.000

Table 8.3: Constraint integrity: the number of rule firings actively enforced per run, with the symbolic layer enabled versus disabled (source: full grid). Because rule compliance is 1.000 in both conditions, it does not distinguish them; constraint integrity, not compliance, is the measure that reveals the work the symbolic layer performs.

Scenario	Constraints enforced (+symbolic)	Constraints enforced (no symbolic)	Compliance
TAX-1	50.3	0	1.000

Beyond enforcement, we find that the symbolic layer renders the simulation auditable. Every change to the world state, that is, every policy activation, compliance penalty, grievance update, and strike, is written to an append-only change log as a structured record of what changed, when, and why. In Figure 8.8, we show this log accumulating over a single run to more than one thousand recorded mutations, together with a breakdown of those mutations by entity type. Because we can inspect each entry individually, the entire trajectory of the simulation can be reconstructed and examined after the fact. We regard this combination of enforced constraints and a complete audit trail as the basis for contribution C2.

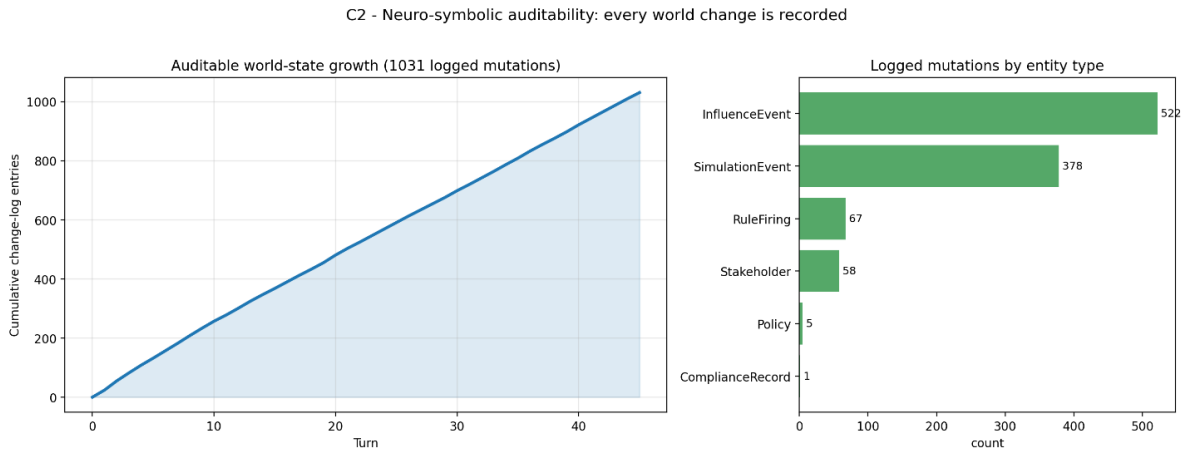


Figure 8.8: Growth of the auditable change log over a single run (left), reaching more than one thousand recorded world-state mutations, with the breakdown of those mutations by entity type (right).

Every entry records what changed, on which turn, and its cause.

In Figure 8.9, we show the knowledge-graph schema that defines the structure this log records. We represent the world state as five entity types, namely Stakeholder, Policy, ComplianceRecord, SimulationEvent, and FinancialState, together with the relations among them. All of the mutations counted in Figure 8.8 are changes to instances of these entities and relations, which allows us to log and audit them in a consistent, structured form.

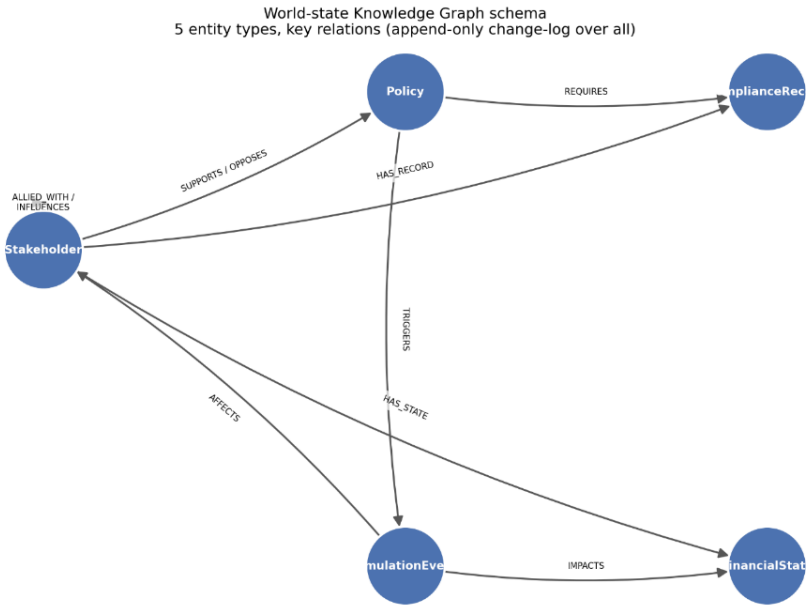


Figure 8.9: Knowledge-graph schema of the world state: five entity types and their principal relations. Every world-state change recorded in the audit log is a mutation of an instance of this schema.

8.4.5 Emotion Dynamics Follow Theory and Produce Stable Opposing Factions

We now examine how the architecture behaves over time and what social structure it produces. We first verify that the emotional substrate behaves as the underlying theory requires. Across all traces, we find that two conformance probes pass without exception: agent arousal rises on shock turns in 100 percent of cases, and remains steady or decays during quiet turns in 100 percent of cases. In Figure 8.10, we make this dynamic visible at the system level. The blue curve is mean agent arousal, which rises at each policy shock and decays during quiet turns; the red curve is total accumulated grievance, which climbs after each shock until it crosses the union mobilization threshold, triggering strikes at turns 36 and 40. We observe that mean arousal across agents (the blue curve) spikes at each policy shock, namely the omnibus announcement at turn 10, the dual-policy rollout at turn 20, and the audit at turn 30, and decays in the intervening quiet periods. At

the same time, accumulated grievance (the red curve) climbs after each shock until it crosses the union mobilization threshold, at which point the union strikes, at turns 36 and 40. In this single figure we therefore demonstrate two mechanisms at once: the theory-conforming dynamics of the emotional substrate, and the symbolic strike mechanism firing as a downstream consequence of accumulated grievance.

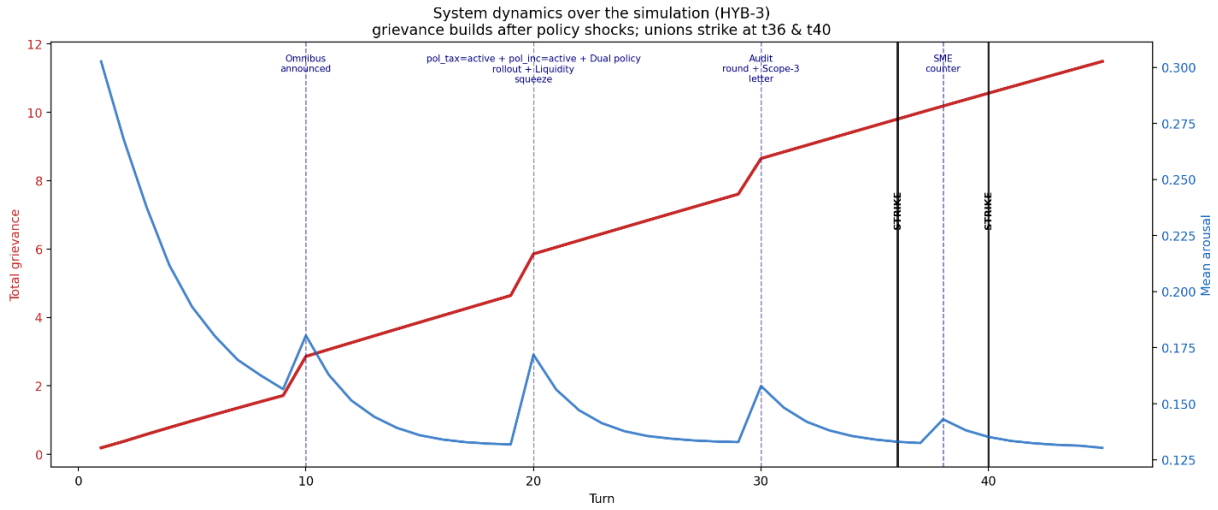


Figure 8.10: System-Level Dynamics Over a Single Run.

In Figure 8.11, we show the same dynamics at the level of individual agents, plotting agent arousal across the full run as a heatmap. We observe that the brightest activity occurs in the early turns and at the shock turns, after which arousal subsides, mirroring the system-level pattern and confirming that the response is distributed across the agent population rather than driven by a few agents.

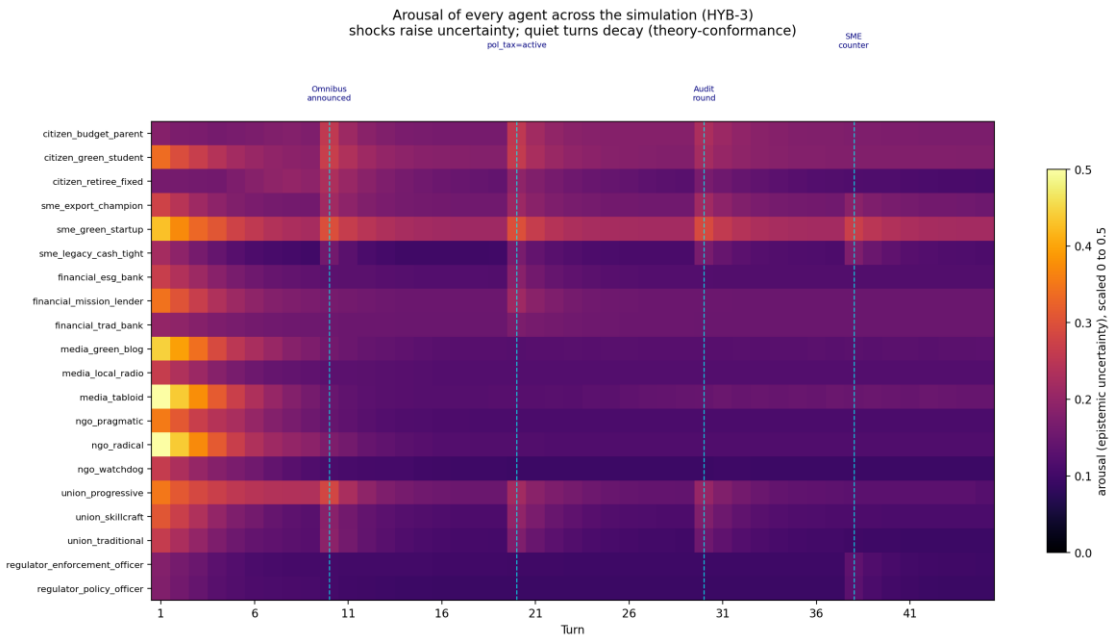


Figure 8.11: Per-agent arousal across the forty-five-turn horizon (hybrid scenario). Brighter cells indicate higher arousal; the vertical structure shows that shocks raise arousal across the population, after which it decays during quiet turns.

We find that these dynamics give rise to a stable, polarized social structure rather than a drift toward consensus. We identify coalitions as agent pairs whose mutual alignment, accumulated over the run, meets or exceeds the 0.50 threshold defined in our influence model. In the representative hybrid run, we find that 62 of the 87 directed agent pairs reach this threshold, and that they organize into two durable groups: a large support bloc of sixteen to seventeen agents and a small, hardened opposition of three to four agents. We observe that the opposition is remarkably persistent: the media tabloid, the radical NGO, and the fixed-income retiree remain opposed for 42 to 45 of the 45 turns. One agent, the budget-conscious citizen, remains genuinely contested throughout, switching between the two camps repeatedly. This pattern is consistent across the grid, where we find dense within-bloc alignment, with 52 to 60 high-alignment edges per run.

In Figure 8.12, we present the full picture of this structure across the run, showing the stance of every agent on every turn, with supporting agents in green and opposing agents in red. We can directly observe the stable support band, the persistent red opposition rows, and the single flickering swing agent. In Figure 8.13, we trace the same structure at six representative turns,

holding the agents in fixed positions so that we can follow the persistence of the two blocs, and the steady rise of accumulated grievance beneath the stable surface, across the run.

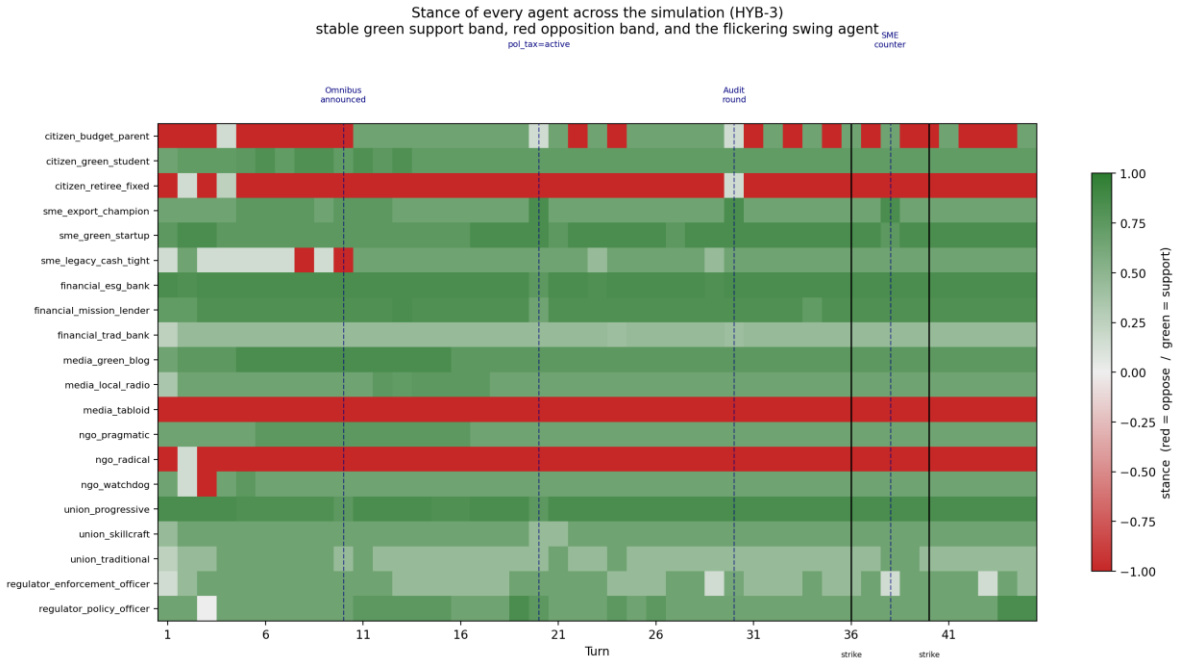


Figure 8.12: Stance of every agent on every turn (hybrid scenario). Green indicates support and red indicates opposition.

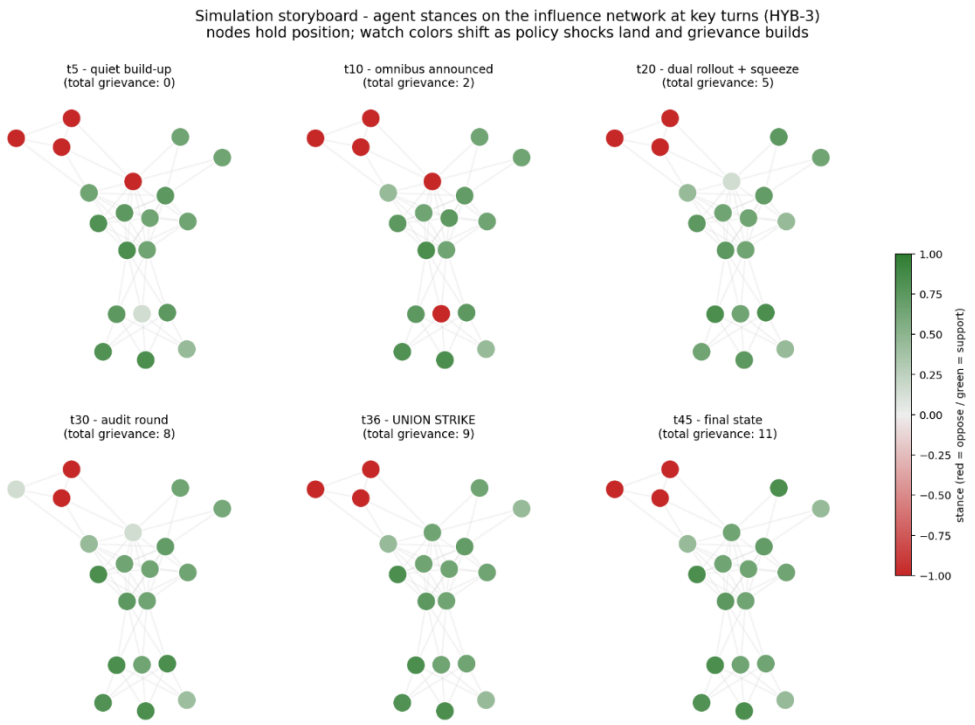


Figure 8.13: Storyboard of the influence network at six representative turns. Agents hold fixed positions across the panels; node color encodes stance and the panel titles report cumulative grievance, showing that the bloc structure remains stable while grievance builds underneath.

Finally, in Figure 8.14 we aggregate the directed influence weights over the run into a class-by-class flow matrix, in order to show which classes shape which others. We observe that the media class is the dominant source of influence, exerting the strongest pull on citizens, unions, and SMEs; that NGOs form a secondary hub; and that regulators and financial institutions act largely as recipients rather than sources of influence. We find this structured flow to be consistent with the influence topology built into the architecture.

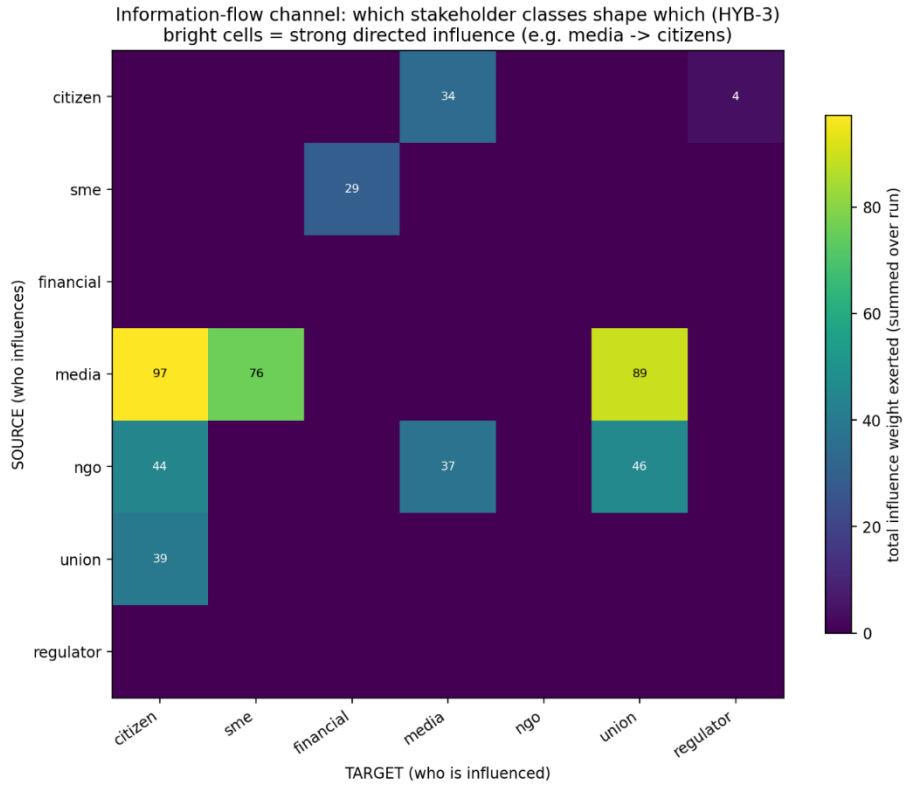


Figure 8.14: Directed information-flow matrix among stakeholder classes, aggregated over the run. Each cell gives the total influence weight a source class (row) exerts on a target class (column); brighter cells indicate stronger influence. Media is the dominant source, with NGOs a secondary hub.

8.4.6 The Findings Hold Up Under Stress Testing

We next establish that the headline findings are not artifacts of incidental choices in the experimental setup. We subject the hybrid scenario to three independent perturbations, which we summarize in Table 8.4. In each row we report the mean and standard deviation of the metric across the perturbation, and from the narrow spreads we read stability. First, we replace the citizen and SME agents with eight alternative persona pairings drawn by Latin hypercube sampling; we find that mean regret remains essentially unchanged at 0.174 plus or minus 0.005, indicating that the results do not depend on the specific agents chosen. Second, we permute the order in which agents act within each layer; we find mean regret tighter still at 0.161 plus or minus 0.001, confirming

that the findings are invariant to action ordering. Third, we sweep three internal calibration parameters across their plausible ranges; we observe that the agents' no-op rate remains flat at 0.189 across every value tested, showing that the architecture does not sit on a knife edge with respect to its parameters. Taken together, we read these checks as establishing that our results are stable with respect to agent identity, action ordering, and parameter calibration.

Table 8.4: Robustness of the hybrid-scenario results under three perturbations (robustness study, recomputed from raw run records).

Perturbation	Mean regret	Stance var.	Note
Persona swap (LHS, 8 pairings)	0.174 +/- 0.005	0.360 +/- 0.047	agent-invariant
Order permutation (3 orderings)	0.161 +/- 0.001	0.403 +/- 0.047	order-invariant
Parameter sweep (3 parameters)	no-op rate flat at 0.189		no knife edge

8.4.7 The Architecture Transfers to a Stronger Model from a Different Family

Finally, we test whether the architecture's behavior depends on the particular language model used to drive the agents. We re-run two conditions on a stronger model from a different model family (GPT-4.1), and, to ensure a fair comparison, we also re-run the primary model (Qwen) under the same constrained decoding format. By holding the decoding format constant in this way, we ensure that the only variable between the two model columns of Table 8.5 is the capability of the underlying model. Our purpose is to separate the aspects of the architecture's behavior that are intrinsic to the design from those that reflect the model.

Table 8.5: Cross-model comparison under matched constrained decoding (source: frontier and matched-baseline runs, recomputed from raw records). Because we hold decoding constant, the only difference between the two model columns is model capability.

Condition	GPT-4.1	Qwen (matched decoding)	Measure
Coherence, HYB-3	0.927	0.314	rho
Coherence, TAX-1	0.913	0.464	rho
Stance variance, HYB-3	0.270	0.214	variance
Stance variance, TAX-1	0.279	0.283	variance
Rule compliance	1.000	1.000	rate

We find that two of the three measures are essentially model-invariant. Rule compliance is identical at 1.000 on both models, from which we confirm that the symbolic layer enforces its constraints regardless of the underlying model, and stance variance lands in the same band on both. On the stronger model, mean regret is also comparable to or slightly lower than on the primary model (0.158 and 0.171 against 0.179 and 0.183 under matched decoding), indicating that the stronger model makes marginally better decisions, as we would expect. Emotion, behavior coherence, by contrast, we find to be model-dependent in magnitude: it is substantially higher on the stronger model (0.92 against 0.39 under matched decoding).

We summarize this study as follows. The architecture itself is portable: its utility across models, and the qualitative effect of each of its levers, namely emotion, rationality, memory, and the symbolic layer, are preserved when we move to a different and stronger model, as evidenced by

the model-invariant compliance, the comparable stance variance, and the consistent direction of every effect. What is *not* preserved is the magnitude of each lever's impact, which depends on the capability of the underlying model, as the coherence values make clear. In other words, we find the architecture and the direction of its mechanisms to be model-independent, while the size of their effect is model-dependent. We regard this distinction as strengthening, rather than weakening, the contributions established above, because those contributions rest on within-model comparisons that we find to hold on both models.

8.4.8 The Same Architecture Produces Very Different Agents

We close the results by illustrating how the same architecture produces qualitatively different agents, depending only on persona rather than on any agent-specific logic. In Figure 8.15, we trace the complete internal state of the swing agent, the budget-conscious citizen identified above, across the run. Reading the panels from top to bottom, we can follow the causal chain of the architecture: the agent's arousal rises at each policy shock, its mood (valence) drifts steadily downward over the run, its decided stance reverses repeatedly between support and opposition, and its regret spikes at exactly those turns where its choices diverge most from the rational baseline. In this figure, we observe the full emotion, memory, and decision loop operating within a single agent.

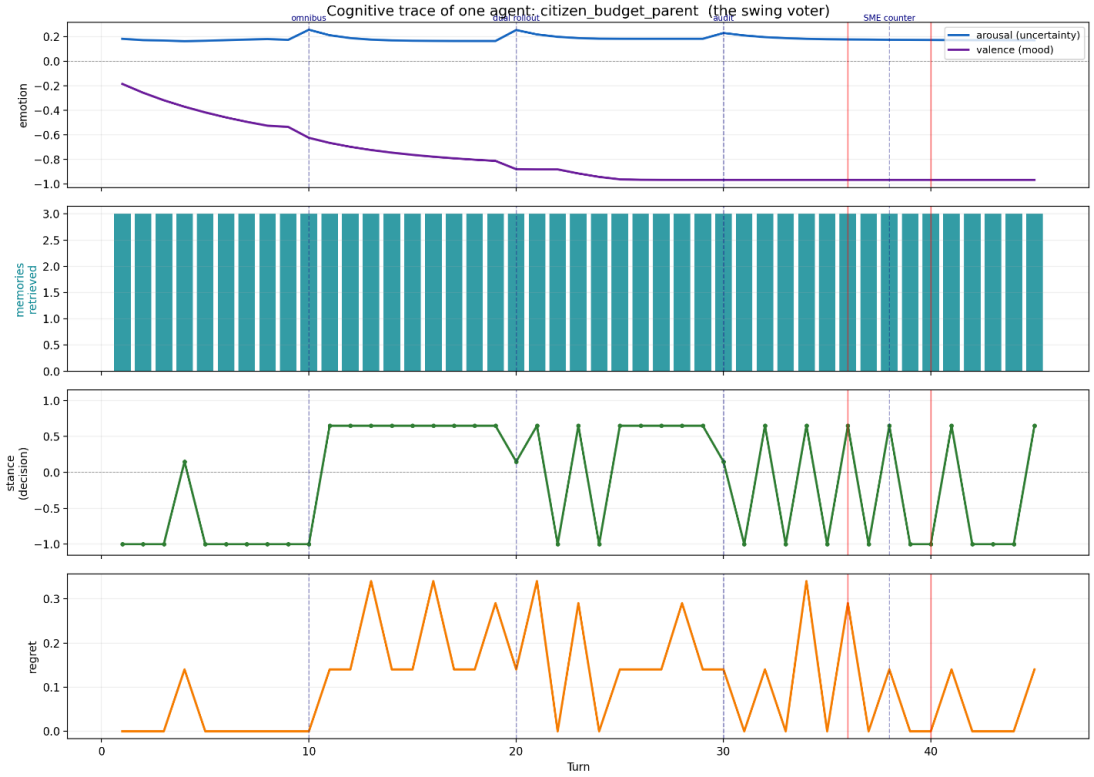


Figure 8.15: Complete cognitive trace of the swing agent (budget-conscious citizen) across the run. From top to bottom: arousal and valence, accumulated grievance, decided stance, and regret. Rising uncertainty and a steadily souring mood accompany the agent's repeated stance reversals.

In Figure 8.16, we trace a second agent, the progressive union, whose behavior follows an entirely different logic on the same architecture. Here we direct attention to the second panel: accumulated grievance rises in distinct steps, each aligned with a policy shock, until it crosses the mobilization threshold and the union strikes at turns 36 and 40. We note that the union maintains a supportive stance throughout (third panel) even as it strikes, which captures a nuanced behavior in which an agent supports a policy in principle while taking collective action over accumulated grievance. The two agents in Figures 8.15 and 8.16 run on the identical emotion, memory, and symbolic loop, and we emphasize that their divergence arises from their personas, not from any agent-specific code, which is our practical demonstration of the architecture's generality.

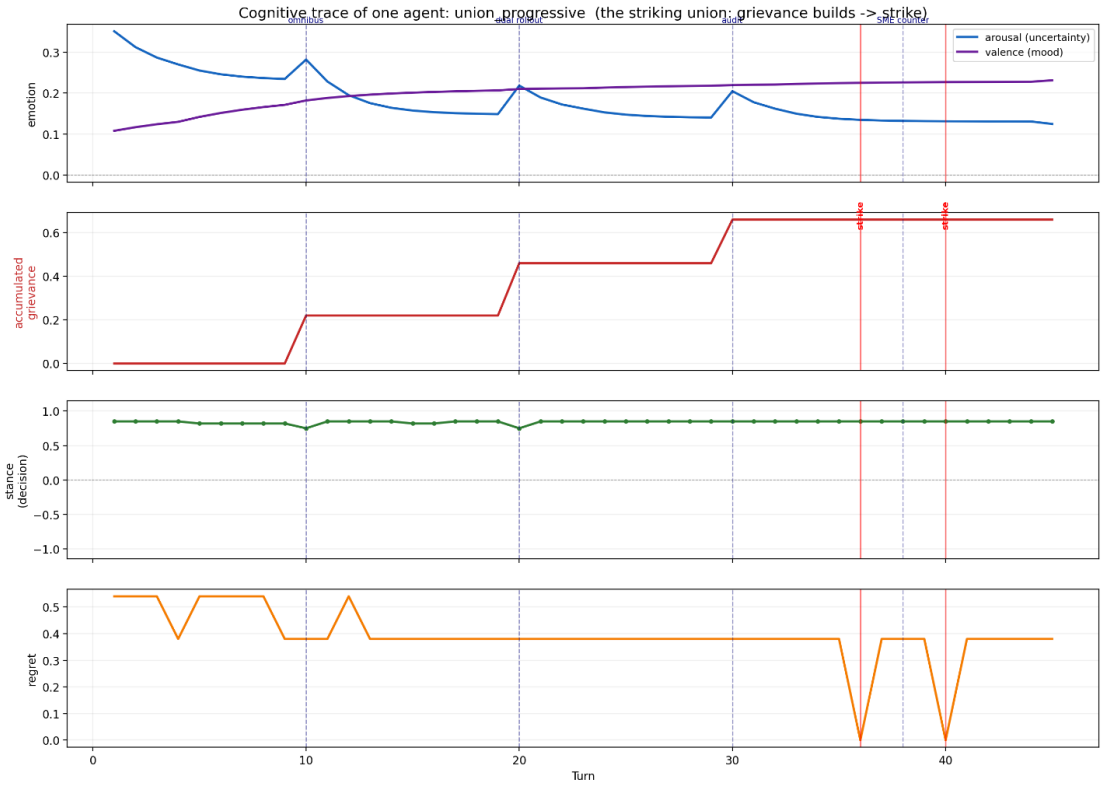


Figure 8.16: Complete cognitive trace of the progressive union across the run. Accumulated grievance (second panel) rises in steps aligned with the policy shocks until it crosses the mobilization threshold, triggering strikes at turns 36 and 40, even as the union maintains a supportive stance (third panel).

8.4.9 Scope of the Reported Evidence

We close with a note on the scope of the evidence, in the interest of precision. The aggregate metrics we report in Tables 8.1 through 8.5 are drawn from the complete 136-run grid, which completed with zero failures, together with the robustness and cross-model studies. The agent-level traces and network visualizations are drawn from a single representative run, and we intend them to illustrate the mechanisms of the architecture rather than to make quantitative claims in their own right. We further note that memory-retrieval precision under our relevance measure is modest. Under a naive token-overlap proxy the precision reads 0.038, while under our independent relevance judge it reads 0.204, roughly five times higher, so the proxy substantially understates true retrieval relevance. We regard the modest precision as a property of the current retrieval

implementation rather than a limitation on the memory contribution itself, since, as we establish in Section 8.4.1, removing memory still produces the largest single degradation in behavior that we observe.

8.5 Discussion

Having reported what the architecture does, we now step back to consider what these results mean, what they cost to obtain, where they are limited, and where they point. We keep this discussion deliberately compact, and we do not repeat the quantitative findings of Section 8.4; instead, we interpret them.

8.5.1 Insights on Demonstration through the Architecture

We set out, in Section 8.1, to make a generative social simulation that could serve as evidence rather than as illustration, and we identified three gaps that stood in the way: emotion that was a label rather than a mechanism, reasoning that could not be audited, and departures from rationality that were described rather than measured. We read the results of Section 8.4 as closing each of these gaps in turn. We find that emotion, expressed through arousal, operates as a genuine inference-control signal: weakening it degrades the consistency of agent behavior along a measured gradient, which means that emotion in our architecture changes how an agent reasons rather than merely how it is labeled. We find that the symbolic layer enforces a substantial body of constraints that a compliance score cannot even detect, and that every change to the world is recorded in a form we can inspect after the fact, which makes the reasoning auditable rather than opaque. And we find that the departure of each stakeholder class from rational behavior is not asserted but measured, as a class-specific quantity of regret against an explicit rational baseline.

Taken together, we read these findings as evidence that the value of the architecture lies in the integration rather than in any single mechanism. From the capability ladder we see that no component carries the result alone: memory contributes the largest share of behavioral coherence, but emotion, the symbolic layer, and the deliberative orchestration each add a measurable and distinct increment, and the architecture behaves as designed only when they operate together. We therefore regard the central demonstration of this work as not that any one of emotion, memory, or symbolic reasoning matters in isolation, which is already known, but that coupling them under a

shared, auditable world produces agents whose behavior is coherent, measurable, and inspectable at the same time.

We close this subsection by placing the work in the context of agentic AI. Much of the current effort in this area is directed at enabling agents to act, that is, to plan, to call tools, and to operate autonomously in extended loops. To that effort we offer a complementary set of tools, aimed not at making agents act but at making what they do measurable and auditable. Three elements of our approach are reusable beyond the setting we study here. The first is a measurement discipline: we evaluate an agent's reasoning through quantities such as constraint integrity and regret rather than through task success or fluent output alone, and our compliance result is a concrete illustration of why surface metrics can mislead, since a compliance score can read as perfect while no constraint is actually in force. The second is an auditable substrate: by separating a deterministic, inspectable mechanism layer from the neural layer, we allow an agent's decisions to be reconstructed and checked after the fact, a design pattern that is not specific to our domain. The third is the treatment of emotion as an inference-control signal rather than as a persona label, a mechanism that others can adopt to make agents respond adaptively to their circumstances. We expect these three elements to be useful to others building agentic systems, whatever their application.

8.5.2 Computational Complexity and Cost

The evidence above is expensive to produce, and we discuss that cost candidly because we believe it is a defining characteristic of this class of work rather than an incidental detail. In our architecture, every agent decision is a language-model call, and a single deliberative cell of twenty agents over forty-five turns requires on the order of 1,350 such calls once the within-turn deliberation rounds are counted. Across the full study, the headline grid, the robustness checks, and the cross-model slice together amount to several hundred thousand agent and judge calls; the independent judge alone made 75,153 scoring calls in the headline campaign. We served the primary agent model locally on a single high-memory GPU using a high-throughput inference engine, which made the per-call cost of the local model negligible in money but not in time, while the frontier and judge models, accessed through a commercial interface, carried a direct monetary cost that scaled with the number of calls.

Several structural pressures follow from this design, and we expect them to apply to any language-model-driven social simulation of comparable ambition, not only to ours. The first is that cost grows multiplicatively, because the number of model calls scales with the product of the agent count, the horizon, and the number of deliberation rounds per turn, so that modest increases in the size or length of a simulation produce large increases in compute.

The second, and in our view the most consequential, is a conflict between reproducibility and throughput that reaches down to the hardware. The natural way to make these simulations fast is to issue many agent calls at once, and a modern accelerator is in fact built to process such calls in large parallel batches. However, when calls are batched together, the numerical result computed for any one agent can depend on which other calls happen to share its batch, because the order of floating-point reductions and the scheduling of work on the device are not fixed across different batch compositions. The practical consequence is that running the same agent on the same seed under concurrent execution can yield a subtly different output from one run to the next. This is the reason we execute every call serially at temperature zero: serial execution removes this source of variability and lets a run be determined by its seed, but it also forgoes the very parallelism the hardware is designed to exploit, so that the accelerator sits largely idle while one call is processed at a time. In this class of work, therefore, reproducibility and speed are not merely a software trade-off but are in tension at the level of the silicon, and the cost we pay is not only that we make many calls but that we are obliged to make them inefficiently.

The third pressure is that the cost of a run is not constant across its length. As a simulation proceeds, each agent accumulates memory, history, and world state, so that the prompt presented to the model grows from turn to turn. Because the cost of language-model inference rises faster than linearly with the length of its input, the later turns of a long-horizon run are individually more expensive than the early ones, and the total cost of a run grows faster than its turn count alone would suggest. Long-horizon agent simulations are thus subject to a scaling pressure that short interactions do not face. To these we add a fourth, smaller point: at the scale of hundreds of thousands of sequential calls, the rare malformed output must be detected and handled gracefully or it will corrupt a run, so that engineering around an occasionally unreliable model substrate becomes a first-class part of the work rather than an afterthought.

We draw a reflection from this that we believe is critical, and that reframes the cost rather than merely lamenting it. The choices that make our study expensive, namely determinism and serial execution, are the same choices that make it reproducible and auditable. A faster, concurrent, and cheaper simulation would not be reproducible at the level of its mechanism, and so it could not serve as the evidence we set out to produce; it would fall back into the category of illustration that we sought to leave behind in Section 8.1. We therefore regard the computational cost not as waste but as the present price of trustworthy simulation. The implication for future work, in our view, is that the goal should not be to make these simulations cheap by abandoning reproducibility, but to lower the cost of reproducible execution, treating auditability and determinism as requirements to be preserved while efficiency improves, rather than as luxuries to be traded away for speed.

8.5.3 Limitations

We note several limitations of the present study, and in each case we read the limitation as an opportunity to extend the work rather than as a boundary on it.

First, we evaluate the architecture on a single domain, the deliberation over carbon policy. The mechanisms we propose are not specific to that domain, but we have not yet shown that the findings transfer to other stakeholder settings such as public-health response or product launch, and establishing that breadth is a natural next step.

Second, the precision of memory retrieval under our relevance measure is modest, which indicates that the agents retrieve more loosely than an ideal system would. We find that memory nonetheless produces the largest single contribution to behavioral coherence, so we read this as headroom in the retrieval mechanism rather than a weakness in the role of memory, and improving retrieval relevance is an opportunity to strengthen an already load-bearing component.

Third, we find that while we preserve the direction of the architecture's effects across language models, the magnitude of one of our measures, emotion–behavior coherence, depends on the capability of the underlying model. This means that we should read the absolute level of coherence as a within-model quantity rather than an absolute one, and it invites a fuller characterization of how the architecture's behavior varies across a wider range of models.

Fourth, we draw our agent-level traces and network visualizations from single representative runs and intend them to illustrate mechanism rather than to support quantitative claims, and our cross-

model comparison covers a deliberately small slice of the design. We regard a broader cross-model and multi-run characterization as a worthwhile extension that would convert these illustrations into aggregate evidence.

8.5.4 Future Research Avenues and Open Research Questions

We see several promising directions for future research, which we group into extensions of scope, of computation, and of data.

(1) **Extension to new domains.** We propose applying the architecture to stakeholder settings beyond carbon policy, such as public-health crisis response, urban planning, and product or pricing decisions, in order to test whether the mechanisms and the measured contributions transfer across domains.

(2) **Stronger memory retrieval.** Given the modest retrieval precision we observe, we propose investigating richer retrieval and relevance models, so that the most load-bearing mechanism in the architecture operates closer to its potential.

(3) **Generalizing the emergent-structure measures.** We propose a fuller treatment of the coalition and influence structure, including alternative thresholds and time-resolved measures of how factions form and harden, in order to characterize the emergent dynamics more completely than a single alignment threshold allows.

(4) **Reducing the computational burden while preserving reproducibility.** Because the cost of these simulations grows multiplicatively with agents, turns, and deliberation rounds, and because the determinism we require forbids the batching that hardware is designed to exploit, we see a clear need for methods that lower the per-decision cost without sacrificing reproducibility. We suggest research into caching and reuse of agent reasoning, distillation of agent behavior into smaller models, and forms of parallel execution that are engineered to remain numerically deterministic, so that throughput can be recovered without giving up the auditability on which this kind of evidence depends. Progress here would directly determine how large and how long a decision-grade simulation can practically be.

(5) **Datasets and ground truth for stakeholder behavior.** A central obstacle to scaling and validating this kind of work is the scarcity of data. Our personas are hand-specified and our agent memories are synthetic, which is sufficient for a controlled study but limits external

validity. We therefore see an important open problem in generating and curating validated datasets of stakeholder behavior, together with ground-truth records against which retrieved memories and predicted reactions can be benchmarked. Building such datasets would allow the community to train, calibrate, and evaluate emotionally and behaviorally grounded agents against real-world behavior rather than against designer intuition.

(6) **Toward interactive and real-time use.** Finally, we ask whether an architecture of this kind can be made responsive enough to support interactive use, in which a decision-maker poses a candidate policy and observes the simulated stakeholder response within a practical time budget. Answering this question depends directly on the computational advances above, and we regard it as the open research question that most directly connects this work to practice.

8.6 Closing Remarks for Chapter 8

In this chapter, we propose a cognitive architecture for language agents that makes a generative stakeholder simulation usable as evidence rather than as illustration. In the architecture, a deterministic emotion and memory substrate governs each agent's internal state, the language model is called only to choose actions conditioned on that state, and a neuro-symbolic layer of a knowledge graph and a rule engine grounds and validates every action against a shared, auditable world. We design the architecture to supply three properties that fluent but ungrounded simulations have lacked: an emotion that operates as a mechanism rather than a label, a reasoning process that can be audited rather than only observed, and a departure from rationality that is measured rather than asserted.

Evaluating this architecture on a conflict-laden carbon-policy testbed, we established four results. We showed that arousal, used as an inference-control signal, makes agents behave more consistently with their internal state; that the symbolic layer enforces a substantial and auditable body of constraints that a compliance rate would conceal; that each stakeholder class departs from a rational reference by a distinct and measured amount of regret; and that, of all the mechanisms in the architecture, structured memory is the most consequential. We further found these behavioral results to be stable under changes of agents, ordering, and calibration, and to transfer in direction, though not always in magnitude, to a stronger language model from a different family. We read

these results together as evidence that the contribution lies in the integration: coupling emotion, memory, and symbolic reasoning under one auditable world yields agents whose behavior is at once coherent, measurable, and inspectable.

We are deliberate about the limits of what we claim. We evaluate the architecture on a literature-grounded synthetic testbed rather than against external records of real stakeholder behavior, and we therefore do not claim calibrated prediction of real-world outcomes; what we demonstrate is the architecture and the auditability, measurability, and mechanistic interpretability it affords. Because this form of evidence is costly to produce, for the reasons we set out in Section 8.5, we take the lowering of that cost while preserving reproducibility to be the central engineering problem this line of work now faces.

Beyond the specific setting we study, our emphasis is complementary to much of the current effort on agentic systems: where that effort is directed at enabling agents to act, our concern has been with making what they do measurable and auditable, and the tools we develop for that purpose are reusable well beyond the deliberation over carbon policy. We see the most valuable next steps as extending the architecture to new stakeholder domains, strengthening its memory retrieval, building the validated datasets of stakeholder behavior that the field still lacks, and making reproducible simulation efficient enough that a decision-maker might pose a candidate policy and observe the simulated response within a practical span of time. We take the methodological groundwork for that goal to be the contribution of this chapter, and we offer it as a step toward stakeholder simulations that a government, a public institution, or a firm could rely on to weigh a decision before it is made.

We opened this chapter by setting out to address RQ4 and to examine Hypothesis 4. Within the scope of the stakeholder-simulation study we present, the evidence is consistent with that hypothesis: grounding the agents' emotion–cognition coupling in predictive-processing principles, with arousal treated as a proxy for epistemic uncertainty that scales how strongly emotional state biases memory retrieval, and grounding all agents in a shared knowledge graph with role constraints enforced through a rule system, produces behavioral simulations that are measurably distinct from rational-actor models and that surface diagnostic information about stakeholder resistance before any intervention is deployed. On that basis we offer the new knowledge we set out to produce: a cognitively grounded stakeholder agent architecture with predictive-processing-

anchored memory–emotion coupling, neurosymbolic knowledge graph grounding, and automated agent generation for pre-deployment social legitimacy assessment. We take this to contribute the Social Assessment layer, the fifth operational layer of the DSIM. We state the scope plainly: these are simulations intended to be diagnostically informative for design, not predictions of how identified individuals will behave. With respect to verification and validation, we read the Frame of Reference as offering theoretical structural validity, the stakeholder-simulation study as a reasonable vehicle for empirical structural validity, and the measured departure from rational-actor behavior, together with the stress and transfer tests, as empirical performance validity within the bounds of this study; theoretical performance validity we defer to Chapter 9.

With the Social Assessment layer in place, all five operational layers of the DSIM have now been established individually: we can read an E-CPSS state from its tri-space signals, structure and adapt decisions within and across states with quantified robustness, augment our judgment when the design space outgrows unaided navigation, execute adaptive strategies autonomously under governance constraints, and assess social legitimacy before deployment. What remains is to show not only that each layer can be built, but that together they constitute a coherent and sufficient architecture for the strategic engineering of an E-CPSS across state transitions. That is the work of Chapter 9, where we assemble the DSIM, map the contributions against the eleven E-CPSS design requirements, and answer the Principal Research Question with the cumulative evidence.

Appendix 8A. Parameters, Biases, Roster, and Scenarios

Table 8A1. Fixed constants and retrieval weights.

Symbol	Meaning	Value
λ	arousal update rate, Equation (8.2)	0.3 default; 0.15 status-quo personas; up to 0.4 composed institutions
$\alpha(a_t)$	affect gain on memory retrieval, Equation (8.4)	$0.5 + 1.0\ a_t$
$\eta(a_t)$	mood update gain, Equation (8.5)	$0.10 + 0.30\ a_t$ (0.10 to 0.40)
γ	loss-aversion multiplier on negative valence, Equation (8.5)	1.0 neutral; 2.0 to 2.25 loss-averse
δ	recovery rate, Equation (8.6)	0.05 default; 0.10 resilient; 0.03 reactive
w_{sem}	semantic weight, Equation (8.4)	0.45
w_{aff}	affect base weight, Equation (8.4)	0.25
w_{rec}	recency weight, Equation (8.4)	0.10
w_{sal}	salience weight, Equation (8.4)	0.10
w_{tag}	tag weight, entities and policy tags, Equation (8.4)	0.10
w_{goal}	goal-congruence weight, Equation (8.4)	0.00 default; 0.15 to 0.20 when confirmation bias is active
w_{conf}	confidence weight, Equation (8.4)	0 (dormant)

Symbol	Meaning	Value
w_{sev}, w_{surp}, w_{vol}	severity, surprise, and volatility weights, Equation (8.3)	0.5, 0.3, 0.2

The five always-on retrieval weights sum to one; the goal-congruence weight w_goal is added on top per persona, and the confidence weight w_conf is present in Equation (4) but held at zero.

Table 8A2. Cognitive biases, their layers, parameters, and justifications.

Bias	Layer	Parameter and value	Justification
Loss aversion	Parameter	$\gamma = 2.0$ to 2.25 on negative-valence memories	Kahneman and Tversky (1992)
Negativity bias	Parameter	$\gamma = 2.0$ on all negative affect	Rozin and Royzman (2001)
Status-quo bias	Parameter and rule	$\lambda = 0.15$, a slow arousal update, with anchoring	change resistance, set ordinally
Recency and availability	Retrieval	w_{rec} raised to 0.25; w_{sal} adjusted	recency and availability effects, calibrated
Confirmation bias	Retrieval	$w_{goal} = 0.15$ to 0.18, with the goal-congruence scorer	myside bias, calibrated
Mission lock	Retrieval and rule	$w_{goal} = 0.20$, with a mission value rule	mandate lock-in, calibrated

Bias	Layer	Parameter and value	Justification
Precedent anchoring	Rule	anchor cap = 0.15, bounding a revision to this fraction of precedent	Tversky and Kahneman (1974)
Ambiguity aversion	Rule	ambiguity penalty = 0.3 on uncertain options	Ellsberg (1961)
Sacred values	Retrieval, rule, and prompt	w_{goal} = 0.20, with a sacred-value rule and a prompt fragment	Tetlock (2003)
Optimism bias	Prompt	a positive baseline mood and a prompt fragment	Sharot (2011)

We express additional persona-level tendencies, such as short-termism, advocacy, and status pride, through these same four layers rather than as separate labels.

Table 8A3. The agent roster and the persona schema.

(a) Roster.

Stakeholder class	Agents	Acting layer
Citizens	3	Primary
Small and medium enterprises	3	Primary
Financial institutions	3	Primary
Media	3	Mediator
Unions	3	Mediator
Non-governmental organizations	3	Mediator

Stakeholder class	Agents	Acting layer
Regulators	2	Institutional
Synthesizer	1	Analysis, outside the roster

(b) Persona schema.

Field	What it sets
Stakeholder class and layer	the agent's class and its acting layer
Goals and their weights	the agent's objectives and their relative importance
Risk tolerance and trust in regulators	the agent's risk attitude and institutional trust
Baseline mood (valence, arousal)	the persona's resting affective state
Affective constants (λ, δ, γ)	the arousal rate, the recovery rate, and the loss-aversion multiplier
Bias configuration	the four-layer settings: retrieval-weight overrides, goal-congruence weight, anchor cap, ambiguity penalty, and susceptibility
Class parameters	class-specific fields: media amplification κ, union threshold θ, solidarity ratio, and lending fields
Seed memories	the agent's affect-tagged starting memories, following the record schema of Section 3.4

Table 8A4. The four scenarios, their policy instruments, and their shocks.

Scenario	Policy instrument	Key parameters	Scheduled shocks
BASE-0	none, an information campaign only, a counterfactual	none	fuel-price rise of 8% at turn 20; viral smog story at turn 35
TAX-1	a flat carbon tax on the small and medium enterprises	$55 per tonne CO2e; 50% first-year rebate	Scope-3 compliance letter at turn 30; SME lobby counter-proposal at turn 38
INC-2	a green-capital incentive	30% capital-expenditure grant up to $300k; 3% concessional loan over seven years	finance liquidity squeeze of 150 basis points at turn 20
HYB-3	the tax and the incentive jointly, a stress test	both instruments	both sets of shocks above

Each run lasts forty-five turns, where one turn stands for about two weeks, and the announcements roll out at fixed turns, for example an announcement at turn 10 and legislation at turn 20.

References

[1] Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S., 2023, "Generative Agents: Interactive Simulacra of Human Behavior," *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22.

[2] Larooij, M., and Törnberg, P., 2025, "Validation Is the Central Challenge for Generative Social Simulation: A Critical Review of LLMs in Agent-Based Modeling," *Artificial Intelligence Review*, 59(1), p. 15.

[3] Carattini, S., Carvalho, M., and Fankhauser, S., 2018, "Overcoming Public Resistance to Carbon Taxes," *Wiley Interdisciplinary Reviews: Climate Change*, 9(5), p. e531.

[4] Douenne, T., and Fabre, A., 2022, "Yellow Vests, Pessimistic Beliefs, and Carbon Tax Aversion," *American Economic Journal: Economic Policy*, 14(1), pp. 81–110.

[5] Simon, H. A., 1955, "A Behavioral Model of Rational Choice," *The Quarterly Journal of Economics*, pp. 99–118.

[6] Sumers, T. R., Yao, S., Narasimhan, K., and Griffiths, T. L., 2023, "Cognitive Architectures for Language Agents," arXiv:2309.02427.

[7] Laird, J. E., Newell, A., and Rosenbloom, P. S., 1987, "Soar: An Architecture for General Intelligence," *Artificial Intelligence*, 33(1), pp. 1–64.

[8] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y., 2004, "An Integrated Theory of the Mind," *Psychological Review*, 111(4), p. 1036.

[9] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y., 2022, "ReAct: Synergizing Reasoning and Acting in Language Models," arXiv:2210.03629.

[10] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S., 2023, "Reflexion: Language Agents with Verbal Reinforcement Learning," *Advances in Neural Information Processing Systems*, 36, pp. 8634–8652.

[11] Russell, J. A., 1980, "A Circumplex Model of Affect," *Journal of Personality and Social Psychology*, 39(6), p. 1161.

[12] Bower, G. H., 1981, "Mood and Memory," *American Psychologist*, 36(2), p. 129.

[13] Marsella, S. C., and Gratch, J., 2009, "EMA: A Process Model of Appraisal Dynamics," *Cognitive Systems Research*, 10(1), pp. 70–90.

[14] Li, C., Wang, J., Zhang, Y., Zhu, K., Hou, W., Lian, J., … and Xie, X., 2023, "Large Language Models Understand and Can Be Enhanced by Emotional Stimuli," arXiv:2307.11760.

[15] Clark, A., 2013, "Whatever Next? Predictive Brains, Situated Agents, and the Future of Cognitive Science," *Behavioral and Brain Sciences*, 36(3), pp. 181–204.

[16] Packer, C., Wooders, S., Lin, K., Fang, V., Patil, S. G., Stoica, I., and Gonzalez, J. E., 2023, "MemGPT: Towards LLMs as Operating Systems," arXiv:2310.08560. doi: 10.48550/arXiv.2310.08560

[17] Garcez, A. D. A., and Lamb, L. C., 2023, "Neurosymbolic AI: The 3rd Wave," *Artificial Intelligence Review*, 56(11), pp. 12387–12406.

[18] Tversky, A., and Kahneman, D., 1992, "Advances in Prospect Theory: Cumulative Representation of Uncertainty," *Journal of Risk and Uncertainty*, 5(4), pp. 297–323.

[19] DeGroot, M. H., 1974, "Reaching a Consensus," *Journal of the American Statistical Association*, 69(345), pp. 118–121.

[20] Kerr, C., and Siegel, A., 1934, "The Interindustry Propensity to Strike: An International Comparison," *Industrial Conflict*, A. Kornhauser, R. Dubin, and A. M. Ross, eds., McGraw-Hill, New York.

[21] Kelly, J., 2012, *Rethinking Industrial Relations: Mobilisation, Collectivism and Long Waves*, Routledge.

CHAPTER 9

Synthesis and Closure

9.1 Objectives in this Chapter

In Chapters 3 through 8, we develop six methods. Through each, we address one escalating design challenge, answer one research question, and demonstrate the result on a test problem drawn from a real design situation for an E-CPSS. Taken chapter by chapter, these are six contributions to the Scholarship of Discovery. What we do not yet establish is that they constitute one framework rather than six methods that share a vocabulary. We establish that in this chapter.

We do three things here. We assemble, by defining the Decision-State Information Model formally as the six-tuple {S, I, D, M, K, T}, grounding each component in the chapter in which we produce it, and arguing why these five layers are critical. We close by weighing the assembled framework against the eleven E-CPSS design requirements we establish in Chapter 2, restating every research question and hypothesis and answering it with the cumulative evidence, establishing the validity of the work beyond the six test problems on which we develop it, and answering the Principal Research Question. We reflect, by stating what we contribute, where the honest edges of that contribution lie, what research follows from those edges, and what the work has meant.

We introduce no new methods, no new mathematics, and no new results in this chapter. Everything we present here we build from what we establish in the preceding eight chapters.

9.2 A Summary of the Dissertation

The problem we address. Within the classical paradigm of engineering design, we work effectively when requirements remain stable, when operating conditions remain reasonably predictable, and when we judge performance against specifications established before deployment. We encounter none of these conditions in an Evolving Cyber-Physical-Social System. We observe continuous reshaping among its cyber, physical, and social spaces, we hold different information at one state than at the next, the objectives that matter at one state are not necessarily the objectives that matter at the next, and we alter the conditions for which we design an intervention by

deploying it. We identify two structural sources of this difficulty. Through information asymmetry, we receive from the system more heterogeneous signals than we organize into actionable intelligence, and we are left data-rich and interpretation-poor. Through intelligence asymmetry, the analytical and cognitive capacity we bring is less than the design decisions require, and we face a wider shortfall at higher orders of system complexity rather than a narrower one at higher levels of individual capability. We do not resolve either asymmetry by resolving the other. A designer who sees the system clearly and cannot reason about it well is as constrained as a designer who reasons powerfully in the wrong space. The problem we address in this dissertation is the design of a coherent, state-indexed framework through which a designer resolves both asymmetries, at every state of the E-CPSS and across every transition between states.

In what follows, we traverse the dissertation chapter by chapter. For each research chapter we state the design challenge we address, the method we develop, the test problem on which we demonstrate it, the DSIM layer and formal component we construct, and the knowledge we produce and carry forward.

Chapter 1. We establish the intellectual commitments in this dissertation. We define the E-CPSS as the design object, adopt Strategic Engineering as the paradigm through which we design and manage technological systems to create and maintain value across uncertain, evolving futures, identify information asymmetry and intelligence asymmetry as the root sources of difficulty, and present six escalating design challenges through which those asymmetries appear as practical problems across our test contexts. We introduce the DSIM as the synthesis target, state the Principal Research Question and four secondary research questions with their hypotheses and expected new knowledge, and organize the validation strategy around the Verification and Validation Square. We are explicit about provenance. We do not pre-specify the DSIM and then test it. We develop six methods in response to genuine design challenges, and we reserve for this chapter the work of establishing that they compose.

Chapter 2. We prepare the theoretical ground. We trace the intellectual lineage of the design object from Cyber-Physical Systems through Cyber-Physical-Social Systems to E-CPSS, and we establish why admitting the social space as a first-class design dimension changes what designing means rather than adding a variable to an otherwise familiar problem. We develop the tri-space coevolution model and the cognitive map through which a designer traces how a change

in one space bears on the conditions of decision-making in the others. Using Gero's Function-Behavior-Structure framework, we classify what an E-CPSS requires of a designer from functional, behavioral, and structural perspectives, and in Table 2.3 we identify eleven characteristics an E-CPSS must possess. We map seven research gaps to the Principal Research Question and to each secondary question, so that we give every question in this dissertation a named and justified origin. The eleven characteristics of Table 2.3 are the standard against which we weigh the assembled framework in Section 9.5.

Chapter 3. The first challenge is observation. Before we design any intervention, we characterize the current state of the E-CPSS across all three spaces simultaneously. The difficulty lies not in access to data but in the absence of a structuring principle through which we map available signals onto a design-relevant representation of the system. Without such a principle, we either reduce the data to the variables familiar from prior work and lose the cross-space causal structure, or we treat it as an undifferentiated feature pool and recover statistical patterns that are accurate and uninterpretable for design purposes. We develop a CPSS-structured predictive modeling method in which we assign observational variables to the cyber, physical, and social spaces before model training, build a random forest regression on that structured representation, and attribute importance through SHAP mapped back to the three spaces after training. We demonstrate the method on the artificial light at night and bird strikes test problem, integrating aviation strike records, light pollution measurements, and migration density estimates across 72,545 records. We construct Layer 1 of the DSIM, the Observation Layer, and with it the component I(si), the structured tri-space characterization of the E-CPSS at state si. The knowledge we produce is predictive. We carry forward a model and a set of tri-space attributed importance values through which we account for the role of each feature in the emergent behavior of the system. We also establish empirically that artificial light at night contributes significantly to bird strike incidence, with light pollution level the strongest predictor among those we examine.

Chapter 4. The second challenge is decision structuring. Once we characterize the state, we face the decisions to be made within it, and in an E-CPSS those decisions are not flat. We take a national agricultural subsidy decision at a macro level, a regional technology deployment decision at a meso level, and an individual farm adoption decision at a micro level. The levels are conditioned on one another, since we shape the incentive structures for meso implementation through the macro choices we make, and we set the conditions for micro decisions through the

meso implementation we choose. Where we treat all decisions as equivalent, or address a single level, we obtain interventions that are internally inconsistent or blind to coupling effects. We develop the Integrated Policy Designing Decision Model, a coupled compromise Decision Support Problem formulation in which we encode decision types, interaction strengths, and hierarchical levels across the three scales, and in which we enforce robustness explicitly through Design Capability Index constraints rather than verify it after the fact. We demonstrate the method on the Agriculture 5.0 test problem, exercising the formulation across sixty-one weight scenarios in DSIDES and visualizing the design space through iSOM component planes. We construct the intra-state half of Layer 2, the Decision Structuring Layer, and with it the component D(si). The knowledge we produce is structural. We carry forward a decision map organized by type, level, and coupling strength, together with DCI values that characterize the robustness of each candidate solution across the range of stakeholder priorities we may encounter.

Chapter 5. The third challenge is adaptability across state transitions. Designing robustly within a state is necessary and insufficient, because an E-CPSS moves, and an intervention that is sound at si typically requires rederivation for si+1. We design not only the intervention for the state we occupy but the principled process through which we obtain the next one from the outcomes and the knowledge we produce at the current state. This requires that we work backward from the state we intend to reach rather than forward from the state we occupy. We develop the Goal-Oriented Inverse Design Method, in which we begin from the desired future state, derive the transitions that render it achievable, enforce DCI constraints at every state transition rather than at the endpoint alone, and update goals through feedback when observed performance departs from projection. We demonstrate the method on the United Arab Emirates sustainable development test problem across three temporal states, beginning the exploration at the 2030 target state, working back through the 2023 transition state to the 2017 current state, and exercising forty weight scenarios to represent the ways in which different stakeholders approach the same problem. We complete Layer 2 and construct the component D(si to si+1), the inter-state decision map. The knowledge we produce is dynamic. We carry forward the inverse pathway from current state to target state, the DCI-enforced conditions attached to each transition, and the trajectories along which feedback revises the goals.

Chapter 6. The fourth challenge is cognitive tractability. At higher orders of E-CPSS complexity we face a correspondingly larger design problem, and where the number of goals, the

dimensionality of the decision space, and the density of trade-offs among goals lie beyond the bounds of what we hold in view, we receive from decision support a set of outputs we cannot navigate meaningfully. We augment judgment without displacing it, since guidance we cannot interrogate is guidance we cannot be responsible for. We develop a three-stage generative AI framework in which a large language model extracts candidate knowledge statements from structured scenario data, we validate each statement quantitatively against scenario outcomes before admitting it to the Knowledge Graph, and the validated graph then supports interactive querying and adaptable solution-space generation when a designer's stated intent proves infeasible. We demonstrate the framework on the hot rod rolling steel manufacturing test problem, across 169 scenarios, six objectives, and two coupled design levels. We construct Layer 3, the Augmentation Layer. The knowledge we produce is generative. We carry forward validated design heuristics that we hold as a queryablequery able Knowledge Graph, which is the same structure on which we draw in Chapters 7 and 8 for purposes we have not yet conceived when we construct it.

Chapter 7. The fifth challenge is autonomous execution. Some E-CPSS change at a pace the human design cycle does not answer. A regulator raises a penalty, and we face three times the cost of inaction. A supplier fails to deliver, and we face five times the cost of delay. The goals appropriate at the start of an operating period are no longer appropriate by its end, and we face the next decision before we reconvene, reformulate, and rerun. When we hand execution to a learning agent, we appear to risk the robustness, adaptivity, and interpretability we establish in the preceding chapters, precisely at the moment of a change in the world on which we train that agent. We carry the satisficing commitment of Chapters 4 and 5 into this regime and make the standard for acceptable performance responsive to the world rather than fixed at design time. We develop the Compiled-Aspiration MDP, in which we treat the Knowledge Graph as a formal state variable alongside the Markov decision process, together with a compiler operator that maps facts to aspirations under stated axioms, and the CA-RS-Q algorithm, in which we hold the channel of value learning apart from the channel of aspiration formation. We establish convergence under a settled graph, a finite-time satisficing rate under a stationary graph, and robustness under graph perturbation bounded by graph edit distance, and we render each decision auditable down to the specific facts that set the aspiration. We demonstrate the framework on the supply chain resilience test problem, where non-stationarity is the defining condition rather than an occasional disturbance. We construct Layer 4, the Execution Layer, contributing to both M(si) and K(si). The

knowledge we produce is experiential. We carry forward the policy experience the agent accumulates as it acts, held in the Knowledge Graph as structured, queryable relationships rather than as weights we cannot inspect.

Chapter 8. The sixth challenge is social legitimacy. A technically excellent intervention that stakeholders reject, resist, or subvert has not succeeded, and we therefore treat legitimacy as a design requirement with measurable consequences rather than as a consideration appended to a completed technical process. Stakeholders respond through perception, memory, emotion, and social influence, which we do not represent through rational-actor models. We find simulations populated by language models difficult to trust as evidence, because we obtain from them emotional behavior stated as a stylistic label rather than as a mechanism, and departures from rational self-interest that are asserted rather than measured. We develop a cognitively grounded stakeholder agent architecture in which we govern the internal state of each agent through a deterministic emotion and memory substrate grounded in a circumplex model of affect over valence and arousal, and in which we take arousal as a proxy for epistemic uncertainty through which we scale the strength of the bias of emotional state on memory retrieval. We call a language model only to choose actions conditioned on that state, and we ground and validate every action through a neuro-symbolic layer of Knowledge Graph and rule engine against a shared, auditable world. We demonstrate the architecture on the carbon emission taxation test problem, instantiating seven stakeholder classes as twenty agents across four scenarios, and we isolate the contribution of each mechanism through a capability ladder that removes one mechanism at a time. We construct Layer 5, the Social Assessment Layer, through which we produce the social legitimacy assessment. The knowledge we produce is simulated. We carry forward a pre-deployment map of stakeholder response patterns, the sources of resistance, and the conditions under which stakeholders extend or withhold legitimacy.

What we establish through this traversal. We observe two properties across the whole of the sequence and in no single chapter of it. The first is that the order is not editorial. We do not structure decisions in a state we have not characterized, we do not derive a pathway across a transition without knowing which decisions connect the states, we do not usefully augment a design process that is not structurally sound, we do not answer a system that outruns us by augmenting a designer who still works at human speed, and we do not treat a strategy as complete before we know how the people it reaches respond to it. The second is that the knowledge

accumulates. By Chapter 8, our characterization of a state draws simultaneously on predictive, structural, dynamic, generative, experiential, and simulated knowledge, none of which is available to us at the outset. In Table 9.1 we summarize the traversal. In Section 9.3 we convert it into a formal definition.

Table 9.1: Chapter-by-Chapter Summary of Design Challenges, Methods, Contributions, and New Knowledge

Chapter	RQ	Design Challenge	Method	Test Problem	DSIM Layer and Component	Knowledge Type K(si)	Requirements Operationalized	New Knowledge (condensed)
1. Motivation and Problem Identification	PRQ	Two asymmetries identified	Strategic Engineering as paradigmatic frame	All six	DSIM named as synthesis target, {S, I, D, M, K, T} stated	Not applicable	Not applicable	Foundation for all questions
2. Critical Literature Review	PRQ	Requirements to be established	Tri-space coevolution model, cognitive map, FBS classification	Water utility illustration	S grounded theoretically	Not applicable	All eleven identified in Table 2.3	Eleven characteristics of an E-CPSS, seven gaps mapped to the research questions
3. Observing E-CPSS	RQ1	Observation	CPSS-structured predictive modeling with SHAP tri-space attribution	ALAN and bird strikes	Layer 1, Observation. I(si)	Predictive. Model and tri-space attributed importance values	Scenario recognition, Context Awareness	CPSS-structured predictive modeling method; ALAN a significant contributor to bird strikes
4. Designing Robust Policies	RQ2a	Decision structuring	Integrated Policy Designing Decision Model,	Agriculture 5.0	Layer 2 intra-state, Decision Structuring. D(si)	Structural. Decision map by type, level, and	Robustness, Resilience	Coupled decision method for robust multi-level

			coupled cDSP with DCI constraints			coupling; DCI values		satisficing strategies under uncertainty
5. Designing Adaptive Strategies	RQ2 b	Adaptability across transitions	Goal-Oriented Inverse Design Method with DCI-enforced transitions and feedback-driven goal updating	UAE sustainable development	Layer 2 inter-state, Adaptive Design. D(si to si+1)	Dynamic. Inverse pathway, transition conditions, goal update trajectories	Self-Adaptability	GOIDM for deriving adaptive strategies across E-CPSS state transitions
6. Augmenting Design Intelligence	RQ3 a	Cognitive tractability	GenAI framework with quantitative knowledge validation and Knowledge Graph, interactive querying, adaptable solution-space generation	Hot rod rolling steel manufacturing	Layer 3, Augmentation. M(si)	Generative. Validated design heuristics as a queryable Knowledge Graph	Scalability, Interoperability, Modularity	GenAI-anchored Knowledge Graph framework for interpretable trade-off navigation
7. Autonomou	RQ3 b	Autonomous execution	Compiled-Aspiration	Supply chain resilience	Layer 4, Execution.	Experientia l. Policy	Self-Adaptability,	Aspiration sourced from

s Strategy Execution			MDP, compiler operator, CA-RS-Q, Theorems 1, 2, 3, and 3-G		M(si) and K(si)	experience held as Knowledge Graph relations	Decentralized Decision Making, Organizability	a dynamically updated Knowledge Graph, carrying convergence, satisficing-rate, and graph-perturbation robustness guarantees, with per-decision auditability
8. Assessing Adaptive Interventions	RQ4	Social legitimacy	Memory-emotion coupled stakeholder agents, arousal as epistemic uncertainty proxy, neuro-symbolic grounding	Carbon emission taxation	Layer 5, Assessment. Social legitimacy assessment supplied to T	Simulated. Response patterns, sources of resistance, conditions of legitimacy	Emergent Behavior, Decentralized Decision Making	Cognitively grounded stakeholder agent architecture for pre-deployment social legitimacy assessment
9. Synthesis and Closure	**PRQ**	**Integration**	**Formal assembly and the requirements argument**	**All six**	**All five layers. {S, I, D, M, K, T} assembled, T established**	**All six types composed through the feed-**	**All eleven**	**The Decision-State Information Model, a**

		forward from K(si) into I(si+1)	**state-indexed design intelligence framework for the strategic engineering of E-CPSS**

We construct Layer 2 across two chapters, in Chapter 4 for the intra-state map D(si) and in Chapter 5 for the inter-state map D(si to si+1). In the requirements column we give the names as they appear in Table 2.3 of Chapter 2. We argue the complete mapping in Section 9.6. We condense each new knowledge statement to one line and reproduce the statements in full in Sections 9.7 and 9.10.

9.3 The Decision-State Information Model, Formal Definition

In Chapter 1 we state the DSIM as the six-tuple {S, I, D, M, K, T} and defer its assembly. We carry out that assembly here, and we present the model as a synthesis we arrive at from the six methods rather than a template we impose on them. We take each component in turn, define it, name the chapter in which we construct it, and identify what a designer holds once it is in hand. We then establish the transition condition, which is the component that belongs to no single chapter and which carries the architectural claim of this dissertation.

We write the model as

$$\mathbf{DSIM = \{S, I, D, M, K, T\}}$$

and we index every component other than S to a state of the E-CPSS. In Figure 9.1 we present the architecture of the assembled framework, in which we show the five operational layers, the six formal components, the transition condition on the advance arc, and the loop through which we feed K(si) forward into I(si+1). For a system that holds still, a designer requires no state index in a design framework, because what they know, decide, and produce is the same at every moment. We index the DSIM because an E-CPSS does not hold still, and because the information available to us, the decisions warranted, the methods appropriate, and the knowledge we produce are all state-dependent.

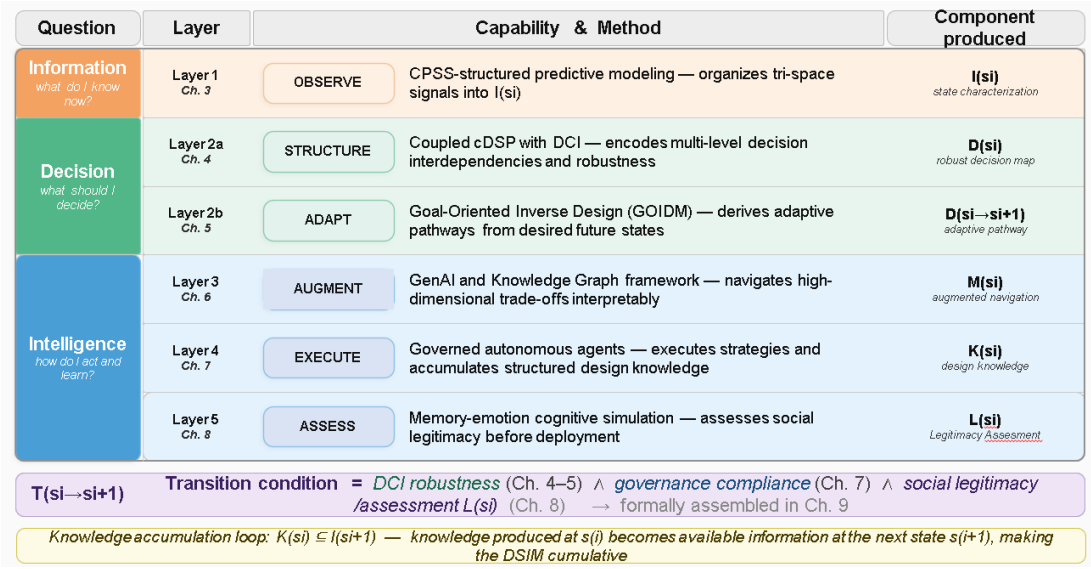


Figure 9.1: Architecture of the Decision State Information Model

9.3.1 S, the set of states

S is the ordered set of states through which an E-CPSS evolves, written S = {s0, s1, ..., sn}. We ground S in the tri-space coevolution model of Chapter 2, where a state of an E-CPSS is a configuration of its cyber, physical, and social spaces together with the coupling relations among them. A state is not a moment in time. We assign two moments to the same state when the conditions under which we design remain those we characterize, and we mark a transition when they no longer are.

We instantiate S separately in each test problem. In the UAE sustainable development problem of Chapter 5, S comprises the current state characterized by historical data through 2017, a transition state at 2023, and the target state at 2030. In the supply chain problem of Chapter 7, the states are the operating conditions we present to the agent as penalties, costs, and supplier reliabilities change within a 52-week horizon. In the carbon taxation problem of Chapter 8, the states are the pre-deployment and post-announcement conditions across which we observe stakeholder positions forming.

In this dissertation, a designer specifies S. We do not derive it from data, and we do not detect the moment at which one state becomes the next. In every test problem, we take the states from the structure of the design situation, whether from a planning horizon with published milestones, from an operating cycle, or from the announcement of a policy. We name this in Section 9.11 and take it up in Section 9.12.

9.3.2 I(si), the structured tri-space characterization

I(si) is the structured tri-space characterization of the E-CPSS at state si. We construct I(si) through the CPSS-structured predictive modeling method of Chapter 3, in which we assign observational variables to the cyber, physical, and social spaces before model training and attribute importance back to those spaces afterward.

I(si) is more than a data set and more than a model. It is our answer to the first of the three questions a designer faces at every state, which is what we know about the system right now. A designer holding $I(s_i)$ holds a representation in which the signals of the system are organized by their space of origin, in which the contribution of each signal to the emergent behavior is quantified, and in which the cross-space causal structure is preserved rather than reduced to an

undifferentiated feature pool. We rely on that last property when we pass I(si) to the layer above it. In a decision formulation we distinguish the variables that are ours to set, the consequences we predict, and the quantities in a space we influence only indirectly. We draw all three distinctions from the tri-space organization.

I(si) is also where we return through the learning loop. We do not assume that the information we hold at a state is complete. We design under that incompleteness through two commitments we carry from the outset, scenario-based exploration, through which we design across the range of conditions the system may present rather than a single assumed one, and satisficing, through which we seek a solution good enough across that range precisely because we do not claim the information to identify an optimum. We return to this in Section 9.3.5.

9.3.3 D(si) and D(si to si+1), the decision maps

D is the decision component, and we build it in two parts because a designer of an E-CPSS holds two distinct decision responsibilities.

D(si) is the intra-state decision map. We construct it through the Integrated Policy Designing Decision Model of Chapter 4, in which we formulate the interventions available at state si as coupled compromise Decision Support Problems that encode decision type, interaction strength, and hierarchical level across macro, meso, and micro scales, and in which we enforce robustness through Design Capability Index constraints as part of the formulation. A designer holding D(si) holds the decisions warranted at si, organized by level, with the couplings among them explicit and with the robustness of each candidate solution quantified across the range of stakeholder priorities we may encounter.

D(si to si+1) is the inter-state decision map. We construct it through the Goal-Oriented Inverse Design Method of Chapter 5, in which we begin from the desired state si+1, derive the transitions that render it achievable, enforce DCI constraints at every transition, and revise goals through feedback when observed performance departs from projection. A designer holding D(si to si+1) holds the pathway out of the current state, with the robustness of each step characterized and with a mechanism through which they update the destination when we observe that the destination was wrong.

We answer different questions through the two parts, and neither substitutes for the other. Through D(si) we address what to do now. Through D(si to si+1) we address how to get from here to where we intend to be. A designer who holds only the first designs well for a state and arrives at the next one without a strategy. A designer who holds only the second holds a trajectory with no account of the decisions that populate any point along it.

9.3.4 M(si), the method set

M(si) is the set of methods we activate at state si. We give this component a property that the others do not hold, since we vary its contents with the complexity of the state rather than with the state alone. We escalate $M(s_i)$ as we find the methods that were adequate to the design problem no longer sufficient for it.

At a state where the decisions are tractable to structured decision support, M(si) contains the compromise DSP constructs, IPDDM, and GOIDM, of Chapters 4 and 5, and a designer formulates, solves, and interprets without further assistance. At a state where the dimensionality of the problem exceeds what we navigate unaided, we add the generative AI and Knowledge Graph framework of Chapter 6, and a designer queries validated heuristics and recovers from infeasible intent rather than examining low-dimensional projections of a space that has more structure than the projections reveal. At a state where we cannot match the pace of the system through the design cycle, we add the Compiled-Aspiration MDP and CA-RS-Q of Chapter 7, and we recompute the standard for acceptable performance from the world rather than fix it when we formulate the problem.

Two observations about this escalation matter for the argument that follows. The first is that the additions are cumulative rather than substitutive. We do not replace the cDSP with generative augmentation, because through the augmentation a designer navigates a satisficing space that we define through the cDSP formulation. We do not replace either with an autonomous agent, because we compile the agent's aspiration from a graph whose contents we obtain through the scenario exploration of the earlier methods. The second is that the escalation is a response to the intelligence asymmetry of Chapter 1 rather than a preference for more elaborate machinery. We add a method where we face the asymmetry at a state, and we argue this in Section 9.4.

9.3.5 K(si), the knowledge we produce and carry forward

K(si) is the knowledge we produce at state si and carry into si+1. Across the six research chapters we produce six types of it. Through Chapter 3 we produce predictive knowledge, comprising a model and tri-space attributed importance values. Through Chapter 4 we produce structural knowledge, comprising a decision map organized by type, level, and coupling, together with DCI values. Through Chapter 5 we produce dynamic knowledge, comprising the inverse pathway, the transition conditions, and the goal update trajectories. Through Chapter 6 we produce generative knowledge, comprising validated design heuristics held as a queryable Knowledge Graph. Through Chapter 7 we produce experiential knowledge, comprising the policy experience the agent accumulates, held in the graph as structured relationships. Through Chapter 8 we produce simulated knowledge, comprising a map of stakeholder response patterns, the sources of resistance, and the conditions under which stakeholders extend or withhold legitimacy.

We tag each item of K(si) to the state at which we produce it, and we feed it into I(si+1). This is the learning loop, and through it we characterize a later state more richly than an earlier one. At s0 a designer characterizes the system from its signals alone. At s1 we draw the same characterization from the signals and from everything we learn at the previous state about which signals matter, which decisions are coupled across the levels of the system, which pathway we are on, which heuristics are supported, what we obtain from the agent as it acts, and where stakeholders resist.

Two properties of the loop are worth stating plainly, because we depend on both later. We hold the knowledge in structured rather than tacit form, and we therefore carry it forward mechanically rather than aspirationally. And the loop belongs to the framework rather than to any method inside it, because in no chapter of this dissertation do we both produce K(si) and take it up as I(si+1).

9.3.6 T(si to si+1), the transition condition

T(si to si+1) is the condition under which a designer authorizes the advance of the E-CPSS from si to si+1. We evaluate it as the conjunction of four criteria, and we assemble it here because we establish it in no single chapter of this dissertation.

We state T as satisfied when all four of the following hold simultaneously.

Robustness. The Design Capability Index values of the designed solution meet or exceed the threshold a designer states, across the range of stakeholder priority configurations under consideration. We obtain this criterion from Layer 2, through Chapters 4 and 5.

Knowledge retention. K(si) is validated and committed to the Knowledge Graph, and available to I(si+1). We obtain this criterion from Layers 3 and 4, through Chapters 6 and 7.

Aspiration satisfaction. The executing agent meets the aspiration we compile from the current graph, so that the state we believe we have reached is the state we occupy. We obtain this criterion from Layer 4, through Chapter 7.

Social legitimacy. The social legitimacy assessment we produce clears the standard a designer states, and the sources of resistance we locate are ones a designer is prepared to answer. We obtain this criterion from Layer 5, through Chapter 8. This is the point at which we bring Layer 5 into the formal model, and it is the only layer through which we contribute to the tuple without constructing a lettered component.

Why we require all four jointly. We treat the four criteria as independent in the sense that matters, which is that we draw no implication about the fourth from the satisfaction of any three. Each is a property of a different object, we measure each in a different space, and we obtain each from a different layer through a different instrument. Robustness is a property of a designed solution under priority uncertainty. Knowledge retention is a property of what we commit. Aspiration satisfaction is a property of realized execution. Social legitimacy is a property of stakeholder response. From a measurement of one of these we obtain no reading on any other.

We see the independence most clearly in the cases where three hold and one fails, and in each such case we describe a failure a designer recognizes.

A designer who establishes technical robustness, retains the knowledge, and clears the legitimacy assessment, but whose executing agent does not meet its aspiration, characterizes si+1 as a state the system has not in fact reached. We take every decision downstream on a state description that is wrong.

A designer who establishes technical robustness, executes to aspiration, and clears the legitimacy assessment, but does not retain the knowledge we produce at si, begins the next state

without the design intelligence we create at the current state. We break the loop of Section 9.3.5, and we are left with a sequence of independent design exercises that happen to occur in order.

A designer who retains the knowledge, executes to aspiration, and clears the legitimacy assessment, but whose solution is not robust across priority configurations, advances on a strategy that we support only for the priority configuration assumed and that we cannot support once priorities change. We encounter changed priorities across transitions, which is when we need the strategy.

A designer who establishes technical robustness, retains the knowledge, and meets the aspiration targets, but cannot secure social legitimacy, advances a strategy against which stakeholders raise resistance sufficient to nullify everything we design.

We evaluate T as a conjunction rather than as a weighted aggregate, and the distinction is not a formality. Under an aggregate, we allow a high reading on one criterion to compensate for a low reading on another, and we admit a technically excellent and socially rejected strategy on the strength of its DCI values. We treat compensation across these four as meaningless, because they are not four measurements of one underlying quantity. They are four separate conditions on whether an advance is sound, and where an advance fails any one of them we hold it unsound in a way that excellence elsewhere does not repair. This follows from how we understand an E-CPSS in Chapter 2, as a system in which the technical, epistemic, institutional, and social dimensions are irreducibly coupled, and in which we obtain fragile strategies where we treat any one in isolation.

We state one boundary here and return to it in Section 9.11. A designer sets the thresholds inside each criterion. We do not derive them, and in each test problem we take them from the requirements of the design situation. What we establish in this dissertation is the structure of T and the necessity of its conjunctive form. What we leave open is the principled determination of where each threshold sits. We also acknowledge that in some situations a designer may replace, revise, or merge some of these criteria as appropriate, though they should not ignore any of them.

Table 9.2 The Components of the Decision-State Information Model

Component	What it is	Chapter	Construct through which we produce it	What a designer holds
S	The ordered set of states through which the E-CPSS evolves	2	Tri-space coevolution model	The states across which the design problem is indexed
I(si)	The structured tri-space characterization at state si	3	CPSS-structured predictive modeling with SHAP tri-space attribution	A representation of the system organized by space, with the contribution of each signal quantified
D(si)	The intra-state decision map	4	Integrated Policy Designing Decision Model, coupled cDSP with DCI constraints	The decisions warranted at si, by level, with couplings explicit and robustness quantified
D(si to si+1)	The inter-state decision map	5	Goal-Oriented Inverse Design Method	The pathway from si to si+1, with robustness enforced at every transition and goals revisable on feedback
M(si)	The set of methods we activate at si	4, 5, 6, 7	cDSP constructs; GenAI and Knowledge Graph framework; Compiled-	The methods appropriate to the complexity of the state, added

			Aspiration MDP and CA-RS-Q	cumulatively as the asymmetry demands
K(si)	The knowledge we produce at si and carry into si+1	3, 4, 5, 6, 7, 8	All six methods	Six types of knowledge, tagged to the state of production, fed forward into I(si+1)
T(si to si+1)	The condition under which we authorize the advance	9	Conjunction of four criteria drawn from Layers 2, 3, 4, and 5	The answer to whether the system may legitimately advance

We construct T in this chapter. We produce it in no single research chapter of this dissertation, because we obtain each of its four criteria from a different layer, and because we hold the conjunction as a relation among the layers rather than as a property of any one of them.

9.4 Why These Five Layers

We assemble five layers. In this section we establish why the set takes this form. We first fix the sense in which we use the word sufficient. We then return to the two asymmetries of Chapter 1 and identify the layer through which we close each asymmetry at each order of complexity. We then address the relationship between the framework and the methods from which we assemble it.

9.4.1 The sense in which the framework is sufficient

We use the word sufficient in a bounded sense, stated here to govern its later use.

We claim that the five layers are sufficient with respect to a stated standard of two parts. The first is that through the layers we close both the information asymmetry and the intelligence asymmetry of Chapter 1, at every state and across every transition. The second is that through the layers we operationalize the eleven characteristics an E-CPSS must possess, which we establish in

Table 2.3 of Chapter 2. When we write that the DSIM is sufficient, we mean sufficient against these two parts.

We do not claim that no further layer could be added to the DSIM, nor that the eleven characteristics of Chapter 2 exhaust what a designer of every E-CPSS requires. A design framework for a class of systems this broad remains open in principle, and a later characteristic or a later design responsibility would warrant a further layer. Our claim is that against the standard we state, we leave no part of it unaddressed through the five layers. We establish the first part of the standard in Section 9.4.2 and the second in Section 9.6.

9.4.2 The two asymmetries, closed layer by layer

In Chapter 1 we trace the difficulty of designing an E-CPSS to two asymmetries, and we argue that if we resolve either without the other we leave the design problem unresolved. We return to them here, because we take the form of the layer set from them.

Information asymmetry is the gap between the volume and heterogeneity of the signals we receive from an E-CPSS and our capacity to organize those signals into intelligence a designer acts on. We close it first at Layer 1. Through the tri-space characterization we construct in Chapter 3, we convert an undifferentiated flow of signals into a representation organized by the space that produces each signal, with the contribution of each signal to the emergent behavior of the system quantified. We close the gap at Layer 1, at the point where signals enter, and we do not close it there for good. At every layer where we produce knowledge the designer did not hold before, we close it again, at a different point and about a different object.

Intelligence asymmetry is the gap between the analytical and cognitive capacity a designer brings and the capacity we require for the design decisions. We meet it at four distinct orders of complexity, and we close it in stages. This is the reason we build four layers above Layer 1, and at each of those four we close the information asymmetry a second time since through them we produce knowledge about the couplings among decisions, about the heuristics latent in scenario data, about the consequences of acting, and about the response of the people an intervention reaches, none of which the signals of the system disclose on their own.

At the order where the decisions are numerous, multi-level, and coupled, a designer cannot hold the couplings and the robustness of many candidate solutions in view at once. We close this

at Layer 2, through the coupled decision formulation of Chapters 4 and 5, in which we carry the couplings and the DCI robustness explicitly. We close the information asymmetry here as well, since the couplings among decisions are a property of the design situation that no signal reports, and we obtain them only by formulating the decisions and examining how they interact.

At the order where we cannot navigate the problem unaided, where the goals, the dimensionality, and the density of trade-offs lie beyond what a designer holds in view, we face an asymmetry of navigation. We close it at Layer 3, through the augmentation of Chapter 6, where a designer queries validated heuristics and recovers from infeasible intent. We close the information asymmetry here as well, since a heuristic that holds across a body of scenarios is knowledge latent in that data rather than stated by it, and we obtain it only by extracting and validating.

At the order where we cannot match the pace of the system through the human design cycle, we face a temporal asymmetry. A designer reasons at the pace of deliberation, and we owe decisions at the pace of the system. We close it at Layer 4, through the autonomous execution of Chapter 7, where we recompute the standard for acceptable performance from the world and we act through the agent within the interval a designer does not answer in person. We close the information asymmetry here as well, since what an agent learns by acting is knowledge about the consequences of action that observation alone does not supply.

At the order where we take the decision outside the technical domain, a designer obtains from no technical analysis an account of how the people who receive a strategy respond, because they respond through perception, memory, emotion, and social influence. We close it at Layer 5, through the stakeholder assessment of Chapter 8, where we produce an assessment of the social response before deployment. We close the information asymmetry here as well, since no signal the system emits before that intervention exists discloses the response of the people it reaches.

Through the transition condition we close both asymmetries again, across the boundary between states. The signals at si+1 differ from the signals at si, so we face the information asymmetry again at every transition, and through the knowledge retention criterion of T we carry the characterization across the boundary. The decisions warranted at si+1 differ from those warranted at si, so we face the intelligence asymmetry again, and through the robustness, aspiration, and legitimacy criteria of T we establish that we have designed soundly for the state we leave before we advance into the next. We summarize the closure in Table 9.3.

We compose the five layers into a framework rather than a collection, and we establish this through two properties. Both are relations among the layers, and neither is a property of any single method. The first is the learning loop of Section 9.3.5. We produce K(si) at each state and feed it forward into I(si+1), so that we draw the characterization of a later state on the knowledge we produce at the earlier states. At no single layer do we both produce this knowledge and take it up as the characterization of the next state. The second is the transition condition of Section 9.3.6. We assemble T from four criteria, and each comes from a different layer, through Layer 2 for robustness, Layers 3 and 4 for knowledge retention, Layer 4 for aspiration satisfaction, and Layer 5 for social legitimacy. We establish T at no single layer, and we hold it only once all five layers are in place. In the final column of Table 9.3 we record what we cannot evaluate in T with each layer removed, which is that we obtain from each layer a criterion that we recover from no other.

Table 9.3: Closure of the Two Asymmetries, Layer by Layer

Layer	**Chapter**	**Asymmetry we close**	**Order of complexity at which it appears**	**Mechanism through which we close it**	**What fails in T if we remove it**
Layer 1, Observation	3	Information asymmetry	Signals beyond interpretation	Tri-space characterization organized by space, contribution quantified	We hold no structure into which to receive I(si+1); we lose the destination of the learning loop
Layer 2, Decision Structuring and Adaptive Design	4, 5	Information and Intelligence asymmetry	Decisions numerous, multi-level, coupled	Coupled decision formulation with DCI robustness carried explicitly	We hold nothing against which to evaluate the robustness criterion
Layer 3, Augmentation	6	Information and Intelligence asymmetry	Problem beyond unaided navigation	Validated heuristics queried, infeasible intent recovered	We hold no validated structure in which to commit and query the knowledge

Layer 4, Execution	7	Information and Intelligence asymmetry	Pace of the system beyond the design cycle	Standard for performance recomputed from the world, action taken through the agent within the interval	We hold no referent for the aspiration criterion
Layer 5, Assessment	8	Information and Intelligence asymmetry	Decision beyond the technical domain	Social response produced before deployment	We cannot evaluate the legitimacy criterion before the advance
T, the transition	9	Both, across the boundary	Signals and decisions differ at si+1	Conjunction of the four criteria supplied by the layers	Not applicable; we assemble T from the layers

We close the information asymmetry first at Layer 1, where signals enter, and again at every layer above it, each of which produces knowledge about the system that its signals do not disclose. We close the intelligence asymmetry at Layers 2 through 5, at four distinct orders of complexity, and at no two layers do we close it at the same order. Through the transition condition we close both again across the boundary between states.

9.5 The Functionalities We Add

In the preceding section we establish why we build the framework in five layers. In this section we state what a designer does through the five layers. We treat a functionality as the operational service we obtain through a method, stated precisely enough to name what a designer supplies to it and what a designer receives from it.

We fix two distinctions before we proceed, and we hold to both throughout the section. A functionality differs from a capability. In Chapter 1 we describe the progression of the dissertation as a capability escalation, running through observing, structuring, adapting, augmenting, executing, and assessing. Through a capability we name what a designer becomes able to do, at the level of the arc. Through a functionality we name the operational service we obtain from one method, at the level of the interface, with its inputs and outputs specified. We compose functionalities, and the composition is the subject of Section 9.5.2. A functionality also differs from the Function of the Function-Behavior-Structure classification we adopt in Chapter 2. There the function belongs to the E-CPSS as the design object, and through it we name what the system

is for. Here the functionality belongs to the designer, and through it we name what the designer does through the method. We use the FBS classification in Section 9.6, and we keep the two referents apart.

We proceed in three parts. We first state the functionality we obtain from each method on its own. We then state the functionalities we hold only where methods meet. We then state what the functionalities amount to across the framework as a whole.

9.5.1 The functionality of each method

We state each functionality as a service, naming what a designer supplies and what a designer receives.

Through the CPSS-structured predictive modeling method of Chapter 3, a designer supplies the heterogeneous signals an E-CPSS produces and receives a tri-space characterization of the state, in which we quantify the contribution of each signal to the emergent behavior of the system and attribute it to the cyber, physical, or social space that produces it. The service is the conversion of an undifferentiated flow of signals into a representation we take up in a decision formulation.

Through the Integrated Policy Designing Decision Model of Chapter 4, a designer supplies a characterized state and a range of stakeholder priorities and receives a coupled multi-level decision map, in which the decisions warranted at the state are organized by type and by foundational, operational, and social engagement level, the couplings among them are explicit, and the robustness of each candidate solution is quantified through Design Capability Index constraints across the priority range. The service is the structuring of interventions within a state so that a designer holds the decisions, their couplings, and their robustness together.

Through the Goal-Oriented Inverse Design Method of Chapter 5, a designer supplies a desired future state and receives the pathway backward to the present, in which the robustness of each step is enforced through DCI constraints at every transition, and the goals are revisable through feedback when observed performance departs from projection. The service is the derivation of a route from where a designer intends to arrive back to present conditions, rather than a projection forward from them.

Through the generative AI and Knowledge Graph framework of Chapter 6, a designer supplies structured scenario data and a stated design intent and receives, across three stages of knowledge creation, knowledge exploration, and adaptable solution-space generation, a set of validated design heuristics held as a queryable Knowledge Graph, an answer to a query stated in natural language, and a recovery when the stated intent proves infeasible. The service is the navigation of a satisficing space whose dimensionality lies beyond what a designer examines through conventional visualization.

Through the Compiled-Aspiration MDP and the CA-RS-Q algorithm of Chapter 7, a designer supplies a Knowledge Graph and a non-stationary process and receives actions taken at the timescale of the system, in which the standard for acceptable performance is recomputed from the current state of the graph, under the convergence, satisficing-rate, and graph-perturbation robustness guarantees we establish, and auditable down to the facts that set the aspiration for each decision. The service is the execution of strategy within intervals a designer does not answer in person.

Through the cognitively grounded stakeholder agent architecture of Chapter 8, a designer supplies a designed strategy and a stakeholder population and receives the social response before deployment, in which the internal state of each agent is governed by a deterministic emotion and memory substrate, every action is grounded and validated against a shared auditable world, and the sources of resistance are located. The service is the production of a social legitimacy assessment before the strategy leaves the designer's hands.

9.5.2 The functionalities we add through composition

We hold several functionalities only where methods meet. Each is a property of a relation between methods rather than of a method, and we obtain none of them from a single research chapter. We compose a construct across the layers rather than an artifact. We establish a construct in one chapter and instantiate it again in a later chapter for a different domain, and we obtain the functionality from that continuity of construct.

We add state-grounded decision formulation where Layer 1 meets Layer 2. We take the goals and variables of the coupled compromise Decision Support Problem of Chapter 4 from a tri-

space characterization of the kind we establish in Chapter 3 rather than from convention, so that we structure the decisions warranted at the state.

We add heuristic-sourced aspiration where Layer 3 meets Layer 4. The Knowledge Graph we establish in Chapter 6 as the structure through which we hold and query validated knowledge is the parallel (with different details) construct we read through the compiler in Chapter 7 to set the aspiration of an autonomous agent. We instantiate the construct separately in each chapter, for hot rod rolling in Chapter 6 and for supply chain operation in Chapter 7, and we carry forward the Knowledge Graph as the device through which we set a standard for action from structured knowledge.

We add shared-world grounding where Layer 3 meets Layer 5. We establish the neuro-symbolic pairing of Knowledge Graph and rule engine in Chapter 6 as a structure a designer queries, and we instantiate it again in Chapter 8 as the structure against which we ground and validate a population of stakeholder agents. We build a separate instance for each domain, and we carry forward the neuro-symbolic construct as the means through which we hold action against an auditable world.

We add legitimacy-conditioned strategy where Layer 2 meets Layer 5. The strategy we assess for social response in Chapter 8 is a strategy of the kind we design in Chapters 4 and 5. For a legitimacy assessment we require a designed artifact, and we return the assessment to the design of that artifact.

We add legitimacy-gated transition where Layer 2 meets Layer 5 at the boundary between states. Through the inverse design method of Chapter 5 we enforce the robustness condition at a transition, and through the stakeholder assessment of Chapter 8 we obtain the legitimacy condition at the same transition. We draw the transition condition of Section 9.3.6 on both, and we establish it in neither chapter alone.

We add cumulative state characterization across every transition. We characterize I(si+1) more richly than I(si) because we bring K(si) into it, and a characterization we draw on predictive, structural, dynamic, generative, experiential, and simulated knowledge at once belongs to the framework rather than to any method within it.

Each of these is a functionality we hold only once we assemble the layers, and each corresponds to a dependency we establish in Section 9.4.

9.5.3 What the functionalities amount to

Taken together, through the functionalities of Sections 9.5.1 and 9.5.2 a designer acts across the full arc of a state and its transition. At an arbitrary state of an arbitrary E-CPSS, a designer reads the state through Layer 1, structures the decisions within it through Layer 2, derives the pathway out of it through Layer 2, navigates the trade-offs through Layer 3 where unaided navigation is not available, executes through Layer 4 where we cannot match the pace of the system through the design cycle, and obtains the social response through Layer 5 before committing. We note that a designer may not face the challenge of designing an E-CPSS as a whole, but may be concerned with only some of these challenges, each of which we address through a different layer and functionality.

One functionality remains, and we obtain it from none of the six methods. Through the transition condition of Section 9.3.6, a designer determines whether the system may legitimately advance from one state to the next. This functionality is the conjunction of the four criteria we obtain from Layers 2 through 5, and we hold it only across the assembled framework. Through the individual functionalities a designer acts within a state and designs its transition. Through the framework functionality a designer decides whether the transition is warranted. We record the full set in Table 9.4.

Table 9.4: The Functionalities We Add, by Method and by Composition

Layer and Chapter	Method	Functionality we add	What a designer supplies	What a designer receives	Where it composes
Layer 1, Ch 3	CPSS-structured predictive modeling	Tri-space characterization	Heterogeneous signals from three spaces	State characterization with signal contributions quantified and attributed by space	Taken up in Layer 2
Layer 2, Ch 4	Integrated Policy Designing Decision Model	Intra-state decision structuring	Characterized state, stakeholder priority range	Coupled multi-level decision map with DCI robustness quantified	Taken up in Layer 3; robustness to T
Layer 2, Ch 5	Goal-Oriented Inverse Design Method	Inter-state pathway derivation	Desired future state	Backward pathway with DCI enforced at every transition and goals revisable on feedback	Robustness to T at the transition
Layer 3, Ch 6	GenAI and Knowledge Graph framework	High-dimensional trade-off navigation	Structured scenario data, stated design intent	Validated heuristics as a queryable graph, natural-language answers, infeasibility recovery	Taken up in Layer 4; knowledge retention to T
Layer 4, Ch 7	Compiled-Aspiration MDP, CA-RS-Q	Autonomous execution at system timescale	Knowledge Graph, non-stationary process	Actions with aspiration recomputed from the graph, under stated	Aspiration satisfaction to T

				guarantees, auditable to the facts	
Layer 5, Ch 8	Cognitively grounded stakeholder agents	Pre-deployment social legitimacy assessment	Designed strategy, stakeholder population	Social response with sources of resistance located	Social legitimacy to T
Framework	**Assembly of all five layers**	**Authorization of a state advance**	**The outputs of Layers 2 through 5**	**The determination of whether we may legitimately advance the system**	**The conjunction we establish as T**

The individual functionalities let a designer act within a state and design its transition. The composed functionalities arise where layers meet, through the constructs re-instantiated per domain, namely state-grounded decision formulation (Ch 3 into Ch 4), heuristic-sourced aspiration (Ch 6 into Ch 7), shared-world grounding (Ch 6 into Ch 8), legitimacy-conditioned strategy (Ch 4 and Ch 5 into Ch 8), legitimacy-gated transition (Ch 5 and Ch 8 into T), and cumulative state characterization (K(si) into I(si+1)). The framework functionality, the authorization of a state advance, is the conjunction we establish as the transition condition T in Section 9.3.6, and we hold it only across the assembled framework.

9.6 Operationalizing the Eleven E-CPSS Design Requirements

In Chapter 2 we identify eleven characteristics an E-CPSS must possess, and we present them in Table 2.3 organized through Gero's Function-Behavior-Structure framework. We claim in Section 9.4 that the five layers are sufficient with respect to a standard of two parts, and the second part is that the layers operationalize these eleven characteristics. We establish that operationalization here.

We route each mapping through the functionalities of Section 9.5 rather than asserting a direct correspondence between a layer and a requirement. The form of each claim is that through a functionality we add, a designer designs an E-CPSS that exhibits the characteristic. We keep two referents apart, as we note in Section 9.5. The characteristic is a property of the E-CPSS as the design object, in the sense of Chapter 2. The functionality is a service a designer performs through a method. A designer operationalizes a characteristic by exercising a functionality through which we obtain a system exhibiting it.

Most of the eleven we operationalize through more than one method, at more than one order of complexity. We name every chapter in which we operationalize a characteristic rather than assign each characteristic to a single home, because we operationalize a characteristic more thoroughly, not less clearly, where we operationalize it across several layers. For each characteristic we mark the chapter in which we demonstrate it on a test problem, and where we contribute to the same characteristic in a further chapter we name that too. We organize the section through the three FBS categories of Chapter 2, and we record the complete mapping in Table 9.5.

9.6.1 Functional requirements

In Chapter 2 we classify Scenario Recognition, Context Awareness, and Decentralized Decision Making as functional requirements, since through them we define the core purpose of an E-CPSS and its capacity to serve that purpose while adapting.

We operationalize Scenario Recognition across all five layers. In the definition of Table 2.3 we identify scenarios by analyzing data from heterogeneous sources to uncover the patterns by which we recognize specific situations. Through the tri-space characterization of Chapter 3, a designer identifies the patterns that characterize a state, which we demonstrate on the bird strikes

test problem where we identify the conditions under which strike incidence rises. Through the augmentation of Chapter 6, a designer identifies which of many scenarios a design situation belongs to, across the 169 scenarios of the hot rod rolling problem. Through the autonomous execution of Chapter 7, we recognize a change of regime, a raised penalty or a failure of supply, as a distinct scenario under which we revise what counts as acceptable performance. Through the stakeholder assessment of Chapter 8, we read the coalition structure we observe under each of four named policy scenarios. The characteristic spans two senses in Table 2.3, and we operationalize both. In the first, a designer identifies which scenario is occurring by analyzing data for the patterns that indicate it, which is what we do through Chapters 3, 6, 7, and 8. In the second, a designer evaluates multiple possible scenarios and their consequences before committing to a decision, which is what we do through the coupled compromise Decision Support Problem of Chapter 4, across sixty-one weight scenarios on the Agriculture 5.0 problem, and through the Goal-Oriented Inverse Design Method of Chapter 5, across forty weight scenarios on the United Arab Emirates problem. In each we examine the consequences of a scenario for the satisficing solution and take the decision across the range rather than at a single assumed configuration. We mark Scenario Recognition demonstrated across all six research chapters, and it is the only characteristic of Table 2.3 for which every method contributes.

We operationalize Context Awareness across three layers together with the learning loop. In the definition of Table 2.3 we interpret changes in the surrounding environment and update context dynamically as those changes occur. Through the tri-space characterization of Chapter 3, a designer reads the context at a state. Through the Compiled-Aspiration MDP of Chapter 7, we bring context into the formal machinery directly, since we hold the context as the knowledge graph from which we set the agent's aspiration and which we update as facts arrive, and we thereby realize the dynamic context updating of Table 2.3 in an operating method. Through the stakeholder agents of Chapter 8, we interpret context separately for each agent through its own memory and emotional state. We establish the dynamic updating of context across states at the framework level, through the forward feed of K(si) into I(si+1). We mark Context Awareness demonstrated in Chapters 3, 7, and 8, with dynamic cross-state updating established through the learning loop.

We operationalize Decentralized Decision Making across two layers. In the definition of Table 2.3 we empower individual components to make independent decisions from local information. Through the CA-RS-Q algorithm of Chapter 7, we enable each agent to decide from

the local state and the current knowledge graph without waiting on a central authority, demonstrated on the supply chain resilience test problem. Through the stakeholder agent architecture of Chapter 8, we form the position of each agent independently from its own interests, memory, and interpretation of the shared world, demonstrated across twenty agents on the carbon taxation test problem. We mark Decentralized Decision Making demonstrated in Chapters 7 and 8.

9.6.2 Behavioral requirements

In Chapter 2 we classify Self-Adaptability, Organizability, Robustness, Resilience, and Emergent Behavior as behavioral requirements, since they define how an E-CPSS adjusts its behavior to evolve and to maintain performance under disruption.

We operationalize Robustness across three layers. In the definition of Table 2.3 we maintain stable performance under uncertainty and disturbance. Through the Integrated Policy Designing Decision Model of Chapter 4, a designer enforces robustness through Design Capability Index constraints so that we support a solution across the range of stakeholder priority configurations, demonstrated on the Agriculture 5.0 test problem across sixty-one weight scenarios. Through the Goal-Oriented Inverse Design Method of Chapter 5, a designer enforces DCI constraints at every state transition rather than at the endpoint alone, and in Chapter 5 we distinguish robustness as a property of a design at a state from adaptability as a property of the process across states, demonstrated on the UAE test problem. Through the CA-RS-Q algorithm of Chapter 7, we establish robustness formally as graceful degradation of policy value under perturbation of the knowledge graph, which is one of the four guarantees we prove for the method. We mark Robustness demonstrated in Chapters 4, 5, and 7.

We operationalize Resilience across two layers. In the definition of Table 2.3 we recover and continue functioning under disruption. Through the inverse design method of Chapter 5, we maintain performance through DCI constraints at every transition as we carry the system between states, demonstrated on the UAE test problem across three temporal states. Through the CA-RS-Q algorithm of Chapter 7, we obtain a clean recovery once a structural change in the world is withdrawn, since we hold the value channel apart from the aspiration channel, demonstrated on the supply chain test problem. We mark Resilience demonstrated in Chapters 5 and 7.

We operationalize Self-Adaptability across two layers. In the definition of Table 2.3 we adjust behavior, structure, or function autonomously in response to changing conditions. Through the Goal-Oriented Inverse Design Method of Chapter 5, a designer revises goals through feedback when performance departs from projection, demonstrated on the UAE test problem. Through the CA-RS-Q algorithm of Chapter 7, we recompute the standard for acceptable performance from the changing knowledge graph, so that we adapt the agent's behavior as the world changes, demonstrated on the supply chain test problem. We mark Self-Adaptability demonstrated in Chapters 5 and 7.

We operationalize Organizability through the autonomous execution of Chapter 7. In the definition of Table 2.3 we organize the components of a system to achieve a desired function, and in Chapter 2 we treat this as both a behavioral and a structural requirement. Through the CA-RS-Q algorithm, we hold the behavior of each agent within governance limits by the structure of the method rather than by external supervision, so that we organize the collective toward the desired function under constraints we audit. We demonstrate this on the supply chain resilience test problem, and we mark Organizability demonstrated in Chapter 7 in both its behavioral and its structural aspect.

We operationalize Emergent Behavior through the stakeholder assessment of Chapter 8. In the definition of Table 2.3 we obtain complex system-level behavior from the interactions among components rather than from explicit programming. Through the stakeholder agent architecture, we obtain coalitions of support and resistance from the interactions among agents rather than from any rule we specify, and we read stable polarized alliances as the emergent social structure of a run. We demonstrate this on the carbon taxation test problem across four scenarios, and we mark Emergent Behavior demonstrated in Chapter 8.

9.6.3 Structural requirements

In Chapter 2 we classify Organizability, Modularity, Scalability, and Interoperability as structural requirements, since they define how the structure of an E-CPSS evolves and reconfigures to fulfill its goals over time.

We operationalize the structural aspect of Organizability through the same governance-constrained structure of Chapter 7, in which we maintain a coherent organization of the agent team

as conditions change. Taken with its behavioral aspect in Section 9.6.2, we mark Organizability demonstrated in Chapter 7.

We operationalize Scalability through the augmentation of Chapter 6. In the definition of Table 2.3 we manage growing data volumes and expanding requirements while maintaining performance. Through the generative AI and Knowledge Graph framework, a designer navigates a satisficing space whose dimensionality lies beyond what we examine through conventional visualization, and we take this as the scalability of the design process to high-dimensional problems, demonstrated on the hot rod rolling test problem across six objectives and two coupled design levels. We mark Scalability demonstrated in Chapter 6.

We operationalize Interoperability through the augmentation of Chapter 6. In the definition of Table 2.3 we exchange information seamlessly among heterogeneous elements. Through the Knowledge Graph, we hold heterogeneous design knowledge in a common structure in which we query and relate elements, demonstrated on the hot rod rolling test problem. We operationalize Interoperability at the level of the design knowledge rather than among the physical subsystems of a deployed E-CPSS, since we build the DSIM as a design framework rather than as a deployed system, and we mark it demonstrated in Chapter 6 at that level.

We operationalize Modularity across two layers. In the definition of Table 2.3 we decompose a system into components we develop independently and then combine or interconnect. Through the Knowledge Graph of Chapter 6, we hold validated knowledge statements as independent units a designer adds, removes, and recombines without rebuilding the structure around them, demonstrated on the hot rod rolling test problem. Through the compiler axioms of Chapter 7, we specify the interface between the knowledge graph and the learning algorithm so that we develop the two independently and interconnect them through a stated contract, which we establish through the axiomatic specification of the compiler. We mark Modularity demonstrated in Chapter 6 and established in Chapter 7.

9.6.4 The completeness of the mapping

Across the three categories, a designer operationalizes all eleven characteristics of Table 2.3 through the functionalities of the five layers. We demonstrate each of the eleven on at least one test problem. Seven of the eleven we demonstrate across two or more chapters, which is Scenario

Recognition, Context Awareness, Decentralized Decision Making, Self-Adaptability, Robustness, Resilience, and Modularity, and four we demonstrate in one chapter each, which is Organizability, Emergent Behavior, Scalability, and Interoperability. Scenario Recognition we operationalize in all six research chapters, and it is the only characteristic to which every method contributes. Two aspects we establish at the framework level rather than demonstrate on a single test problem, which are the dynamic cross-state updating of Context Awareness, carried by the learning loop, and the method-architecture aspect of Modularity, carried by the compiler axioms. This is the second part of the sufficiency standard of Section 9.4.1, and with it in place we meet the standard in both its parts. We record the full mapping in Table 9.5.

Table 9.5: Operationalization of the Eleven E-CPSS Design Requirements Across the Five Layers

FBS category	Characteristic	Chapters that operationalize it	Functionality through which we operationalize it	Test problems	Status
Functional	Scenario recognition	3 (primary), 4,5, 6, 7, 8	Tri-space characterization; trade-off navigation; autonomous execution; stakeholder assessment	ALAN and bird strikes; Agriculture 5.0, UAE Sustainable Development, hot rod rolling; supply chain; carbon taxation	Demonstrated
Functional	Context Awareness	3 (primary), 7, 8; framework	Tri-space characterization; compiled-aspiration context; agent context interpretation; cumulative state characterization	ALAN and bird strikes; supply chain; carbon taxation	Demonstrated; dynamic cross-state updating Established via the learning loop
Functional	Decentralized Decision Making	7, 8	Autonomous execution; stakeholder assessment	Supply chain; carbon taxation	Demonstrated
Behavioral	Self-Adaptability	5 (primary), 7	Inter-state pathway derivation; autonomous execution	UAE; supply chain	Demonstrated

Behavioral	Organizability	7	Autonomous execution, governance-constrained structure	Supply chain	Demonstrated (behavioral and structural)
Behavioral	Robustness	4 (primary), 5, 7	Intra-state decision structuring with DCI; inter-state DCI at every transition; graph-perturbation robustness guarantee	Agriculture 5.0; UAE; supply chain	Demonstrated
Behavioral	Resilience	5 (primary), 7	Inter-state DCI at every transition; recovery under withdrawn perturbation	UAE; supply chain	Demonstrated
Behavioral	Emergent Behavior	8	Stakeholder assessment	Carbon taxation	Demonstrated
Structural	Organizability	7	Autonomous execution, governance-constrained structure	Supply chain	Demonstrated
Structural	Modularity	6 (primary), 7	Knowledge Graph units; compiler axioms	Hot rod rolling; framework	Demonstrated (Ch 6); Established (Ch 7 method architecture)
Structural	Scalability	6	High-dimensional trade-off navigation	Hot rod rolling	Demonstrated

Structural	Interoperability	6	Common Knowledge Graph structure for heterogeneous knowledge	Hot rod rolling	Demonstrated at the level of design knowledge

We give the characteristic names as they appear in Table 2.3 of Chapter 2. We mark the primary demonstration of each characteristic where indicated, and we name every chapter in which we operationalize it. Organizability appears in both the behavioral and the structural band, following the treatment in Section 3.3 of Chapter 2. We mark a mapping demonstrated where we operationalize the characteristic on a test problem, and established where we operationalize it at the level of the framework.

9.7 Answering the Research Questions and Evaluating the Hypotheses

In Chapter 1 we state a Principal Research Question and four secondary research questions, and we divide two of them into sub-questions. We divide RQ2 because through the Decision Layer we hold two distinct responsibilities, structuring decisions robustly within a state and structuring them adaptively across a transition, and we divide RQ3 because through the Augmentation and Execution Layers we address two distinct design conditions. We therefore answer six questions in total, one for each of the six methods through which we construct the five layers. In this section we answer each of the six, and we reserve the Principal Research Question for Section 9.9, where the assembled framework is the answer.

We treat each of the six the same way. We restate the question, the hypothesis, and the expected new knowledge as we state them in Chapter 1. We then present the evidence in three parts, following the Verification and Validation Square we adopt in Chapter 1 and carry through every research chapter. We read the theoretical structural validity from the soundness of the constructs, the empirical structural validity from the suitability of the test problem, and the empirical performance validity from the results we obtain on the test problem. We reserve theoretical performance validity, the generalization of each method beyond its test problem, for Section 9.8, since we argue it for the framework as a whole rather than method by method. We then state the new knowledge we produce through the question. We record the evaluation of every hypothesis in Table 9.6.

In Figure 9.2 we summarize the validation tasks we carry out across the dissertation, and we annotate each quadrant of the Verification and Validation Square with the content through which we address it. In Chapter 1 we present the same square as the plan we adopt, with the theoretical performance quadrant open and deferred to this chapter. Here we present it closed.

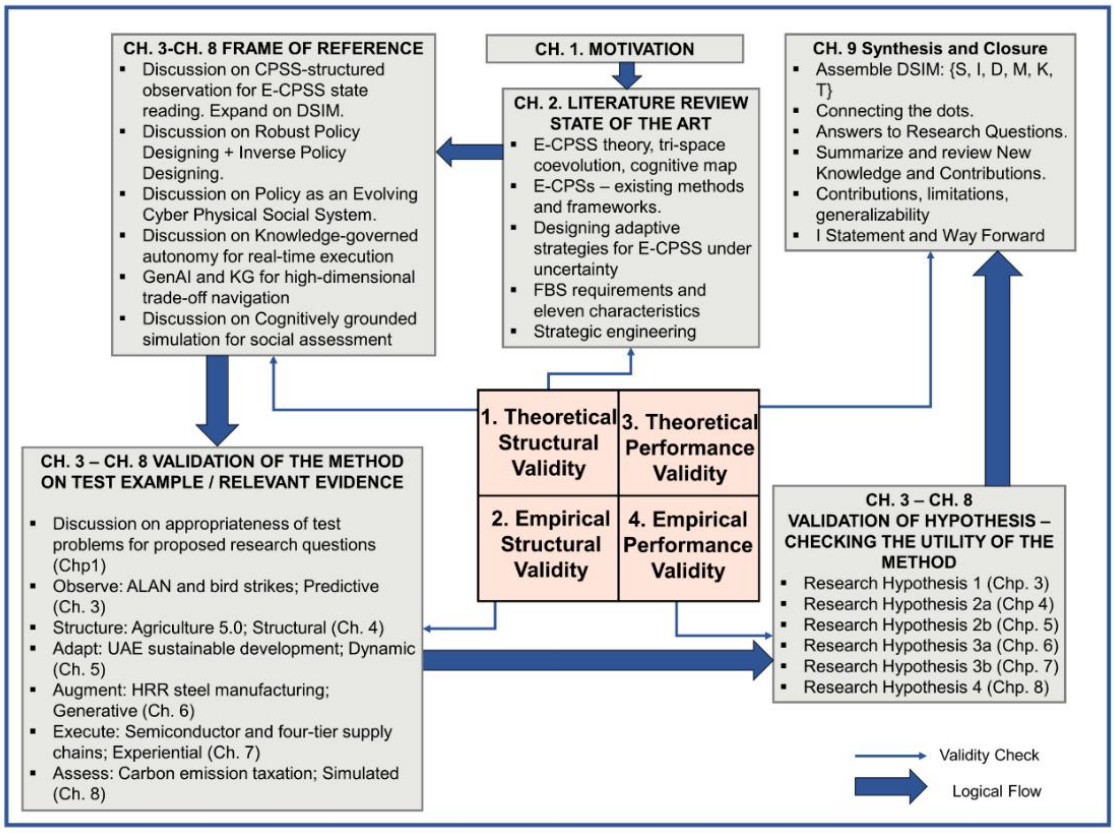


Figure 9.2: Verification and Validation Square

9.7.1 Observation, RQ1

We state RQ1 in Chapter 1 as follows.

> *RQ1. What are the principles and the method for leveraging the cyber-physical-social structure of an E-CPSS in predictive modeling to uncover contributing factors and support design decisions?*

We state Hypothesis 1 as follows.

> *Hypothesis 1. Through a CPSS-structured feature organization that assigns observational variables to cyber, physical, and social spaces before model training, and SHAP-based importance attribution mapped back to those spaces after training, we construct interpretable tri-space state characterizations that reveal the*

contribution of ALAN to bird strikes and yield actionable insights for public policy and aviation safety decision-making.

We establish the theoretical structural validity of the method in Chapter 3 through the tri-space organizing principle of Chapter 2, on which we ground the assignment of observational variables to the cyber, physical, and social spaces before model construction. We establish the empirical structural validity through the artificial light at night and bird strikes test problem, on which we draw signals from all three spaces, integrating aviation strike records, light pollution measurements, and migration density estimates across 72,545 records, and through which we exercise the tri-space organization on a problem that requires it. We establish the empirical performance validity through the results, where the structured model attains accuracy comparable to an unstructured model while remaining interpretable as a description of the tri-space state, and where we establish that light pollution level is the strongest predictor of bird strike incidence among the factors we consider. The evidence within the scope of this problem is consistent with Hypothesis 1.

The new knowledge we produce we state as follows.

New Knowledge 1. A CPSS-structured predictive modeling method for constructing interpretable tri-space state characterizations of Evolving Cyber-Physical-Social Systems to support design decisions. Empirical confirmation that Artificial Light at Night is a statistically significant contributing factor to bird strikes, with light pollution level identified as the strongest predictor, with implications for public policy and aviation safety.

9.7.2 Decision structuring, RQ2a

We state RQ2a in Chapter 1 as follows.

RQ2a. What is the mathematics and method for designing robust strategies in an E-CPSS through coupled, multi-level decisions under uncertainty?

We state Hypothesis 2a as follows.

Hypothesis 2a. By modeling E-CPSS interventions as coupled compromise Decision Support Problems that encode decision types, interaction strengths, and

hierarchical levels using the Integrated Policy Designing Decision Model, with Design Capability Index constraints enforcing robustness explicitly, we produce satisficing solutions that remain robust across stakeholder priority configurations under uncertainty.

We establish the theoretical structural validity through the compromise Decision Support Problem construct and the Design Capability Index, on which we ground the formulation of coupled multi-level decisions and the explicit enforcement of robustness. We establish the empirical structural validity through the Agriculture 5.0 test problem, whose macro, meso, and micro decisions across government subsidy policy, regional technology deployment, and individual farmer adoption exercise the coupled multi-level structure we build the method for. We establish the empirical performance validity through the results, where the classification through the IPDDM reveals that three of the six decision variables interact bidirectionally and are better formulated as coupled than as sequential compromise Decision Support Problems, and where the DCI constraints indicate that the satisficing solution space holds across the stakeholder priority scenarios we examine. The evidence within the scope of this problem is consistent with Hypothesis 2a.

The new knowledge we produce we state as follows.

New Knowledge 2a. A coupled decision method anchored in the Integrated Policy Designing Decision Model and Design Capability Indices for designing robust multi-level satisficing strategies in Evolving Cyber-Physical-Social Systems under uncertainty.

9.7.3 Adaptive design, RQ2b

We state RQ2b in Chapter 1 as follows.

RQ2b. What are the principles, mathematics, and method for designing adaptive strategies from desired future states of an E-CPSS under uncertainty?

We state Hypothesis 2b as follows.

Hypothesis 2b. By using the Goal-Oriented Inverse Design Method to work backward from desired future states, enforce Design Capability Index constraints at every state transition, and update goals through feedback when observed

performance deviates from projections, we produce adaptive strategies that remain robust across E-CPSS state transitions under uncertainty.

We establish the theoretical structural validity through the inverse design principle and the distinction we draw between robustness as a property of a design at a state and adaptability as a property of the process across states, together with the DCI-enforced transition conditions and the feedback goal-updating rules. We establish the empirical structural validity through the United Arab Emirates sustainable development test problem, whose three temporal states spanning 2017 through 2030 give us a design horizon long enough to exercise adaptation across transitions rather than at a single endpoint. We establish the empirical performance validity through the results, where working backward from the target state with DCI constraints enforced at each transition, and updating goals through feedback when observed performance departs from projection, yields adaptive strategies that remain robust across the transitions we model, with goal adjustments derived from the observed transition-state performance. The evidence within the scope of this problem is consistent with Hypothesis 2b.

The new knowledge we produce we state as follows.

New Knowledge 2b. A Goal-Oriented Inverse Design Method anchored in the principle of working backward from desired future states, with DCI-enforced transition conditions and feedback-driven goal updating, for deriving adaptive strategies across E-CPSS state transitions under uncertainty.

9.7.4 Augmented intelligence, RQ3a

We state RQ3a in Chapter 1 as follows.

RQ3a. What are the principles and method a designer uses to enable interpretable and adaptive trade-off navigation when E-CPSS design problems exceed human cognitive tractability?

We state Hypothesis 3a as follows.

Hypothesis 3a. By deploying a generative AI framework in which a large language model extracts knowledge statements from structured scenario data, each statement is validated quantitatively before Knowledge Graph ingestion, and the validated

graph supports interactive querying and adaptive infeasibility recovery, we navigate high-dimensional satisficing spaces with interpretable and actionable guidance for design decision-making.

We establish the theoretical structural validity through the three-stage structure of knowledge creation, knowledge exploration, and adaptable solution-space generation, in which we validate quantitatively before ingestion so that a designer receives guidance resting on validated statements rather than on unvalidated generation. We establish the empirical structural validity through the hot rod rolling steel manufacturing test problem, in which 169 scenarios, six objectives, and two coupled design levels give us a satisficing space of a dimensionality beyond unaided navigation, and through which we exercise the augmentation we obtain from the method. We establish the empirical performance validity through the results, where a designer queries validated heuristics, receives guidance stated in natural language, and recovers an adaptable solution space when a stated intent proves infeasible. The evidence within the scope of this problem is consistent with Hypothesis 3a.

The new knowledge we produce we state as follows.

New Knowledge 3a. A GenAI-anchored Knowledge Graph framework for augmenting E-CPSS design decision-making, with LLM-based knowledge extraction, quantitative validation protocols, and interactive trade-off summarization for interpretable guidance in high-dimensional design spaces.

9.7.5 Autonomous execution, RQ3b

We state RQ3b in Chapter 1 as follows.

RQ3b. What are the principles and method for enabling an autonomous agent to execute adaptive strategies in an E-CPSS under non-stationary conditions, while preserving the robustness, adaptivity, and interpretability established in the preceding chapters?

We state Hypothesis 3b as follows.

Hypothesis 3b: An agent that draws its standard for acceptable performance from a structured, dynamically updated description of the world maintains satisficing

performance and adapts more reliably under non-stationary E-CPSS conditions than an agent that learns from the reward signal alone, without surrendering the convergence, before-convergence, robustness, and auditability guarantees a high-stakes setting requires.

We establish the theoretical structural validity through the Compiled-Aspiration MDP and the CA-RS-Q algorithm, in which we hold the channel of value learning apart from the channel of aspiration formation, together with the four theorems we prove, covering convergence under a settled graph, a finite-time satisficing rate under a stationary graph, and robustness of policy value under graph perturbation bounded by graph edit distance. We establish the empirical structural validity through the supply chain resilience test problem, in which non-stationarity is the defining condition rather than an occasional disturbance, so that we exercise exactly the regime we address through the method. We establish the empirical performance validity through the results across the inventory benchmarks, where by compiling the aspiration from the knowledge graph and holding the two channels apart we enable the agent to follow a structural change in the world and to recover once that change is withdrawn, while we observe the value estimates converging, the actions meeting the aspiration on a meaningful fraction of decisions during learning, and the policy degrading gracefully under graph error, with each decision auditable to the facts from which we set the aspiration. We are careful about the scope of the claim, since in stationary and well-specified regimes the agent is deliberately conservative and we make no claim to an advantage in raw return. The evidence within the scope of these benchmarks is consistent with Hypothesis 3b.

The new knowledge we produce we state as follows.

New Knowledge 3b: A method for sourcing a reinforcement-learning agent's satisficing aspiration from a structured, dynamically updated Knowledge Graph, the Compiled-Aspiration MDP and the CA-RS-Q algorithm which keeps value learning and aspiration formation in separate channels, carries convergence, finite-time satisficing-rate, and graph-perturbation robustness guarantees, and renders each decision auditable to the facts that set the aspiration, for autonomous strategic execution in non-stationary E-CPSS.

9.7.6 Social assessment, RQ4

We state RQ4 in Chapter 1 as follows.

> *RQ4. What are the principles and method for modeling stakeholder behavior through cognitively grounded agent simulation to support pre-deployment social legitimacy assessment of E-CPSS interventions?*

We state Hypothesis 4 as follows.

> *Hypothesis 4. By designing stakeholder agents whose emotional influence on cognition is grounded in predictive-processing principles, specifically by treating arousal as a proxy for epistemic uncertainty that dynamically scales how strongly emotional state biases memory retrieval, and by grounding all agents in a shared knowledge graph with role constraints enforced through a rule system, we produce behavioral simulations that are measurably distinct from rational-actor models and yield diagnostic pre-deployment information about stakeholder resistance before any intervention is deployed.*

We establish the theoretical structural validity through the cognitive architecture, in which we govern the internal state of each agent through a deterministic emotion and memory substrate grounded in a circumplex model of affect, we take arousal as a proxy for epistemic uncertainty through which we scale the strength of the bias of emotional state on memory retrieval, and we ground every action against a shared auditable world through a neuro-symbolic layer of knowledge graph and rule engine. We establish the empirical structural validity through the carbon emission taxation test problem, in which we instantiate seven stakeholder classes as twenty agents holding conflicting interests across four scenarios, so that we exercise the emotional, memory-laden, and socially transmitted response we build the architecture to represent. We establish the empirical performance validity through the capability ladder, in which we remove one mechanism at a time across the experimental grid, and through which we establish that structured memory is the most consequential mechanism for coherent behavior, that arousal coupling measurably improves consistency, that through the symbolic layer we enforce a body of constraints a compliance score would conceal, and that each stakeholder class departs from rationality by a distinct and measured amount, results we find stable under perturbation of the agent population and under a change of

language model. The evidence within the scope of this test problem is consistent with Hypothesis 4.

The new knowledge we produce we state as follows.

> *New Knowledge 4. A cognitively grounded stakeholder agent architecture with predictive-processing-anchored memory-emotion coupling, neurosymbolic knowledge graph grounding, and automated agent generation for pre-deployment social legitimacy assessment of adaptive interventions in Evolving Cyber-Physical-Social Systems.*

9.7.7 The evidence across the six questions

Across the six questions, the evidence within the scope of each test problem is consistent with the hypothesis, and through each question we produce the new knowledge we set out to produce. We evaluate each hypothesis within the bounds of the test problem on which we demonstrate it, and we defer to Section 9.8 the argument that we carry the methods beyond those bounds. We record the evaluation in Table 9.6.

Table 9.6: Evaluation of the Hypotheses Across the Verification and Validation Square

Research Question	Layer	H	Theoretical structural validity	Empirical structural validity	Empirical performance validity	Evaluation
RQ1	Layer 1, Observation	H1	Tri-space organizing principle, SHAP attribution by space	ALAN and bird strikes, draws on all three spaces, 72,545 records	Comparable accuracy with tri-space interpretability; light pollution the strongest predictor.	Consistent within the test problem
RQ2a	Layer 2, Decision Structuring	H2a	Coupled cDSP, Design Capability Index	Agriculture 5.0, macro-meso-micro coupled decisions	Three of six variables coupled; satisficing region holds across priority scenarios.	Consistent within the test problem
RQ2b	Layer 2, Adaptive Design	H2b	Inverse design principle, DCI at every transition, feedback goal updating	UAE sustainable development, three temporal states 2017 to 2030	Adaptive strategies robust across transitions; goals updated from transition-state performance.	Consistent within the test problem
RQ3a	Layer 3, Augmentation	H3a	Three-stage creation, exploration, generation with validation before ingestion	Hot rod rolling, 169 scenarios, six objectives, two levels	Interpretable querying and infeasibility recovery in a high-dimensional space.	Consistent within the test problem
RQ3b	Layer 4, Execution	H3b	Compiled-Aspiration MDP, CA-RS-Q,	Supply chain resilience, non-stationarity as the defining condition	Followed and recovered under structural change; convergence, satisficing	Consistent within the benchmarks

			separated channels, four theorems		rate, graceful degradation, auditability.	
RQ4	Layer 5, Assessment	H4	Emotion-memory substrate, arousal as uncertainty proxy, neuro-symbolic grounding	Carbon taxation, seven classes, twenty agents, four scenarios	Behavior measurably distinct from rational-actor; memory most consequential; per-class departures measured.	Consistent within the test problem

We evaluate each hypothesis within the bounds of the test problem on which we demonstrate it, reading theoretical structural, empirical structural, and empirical performance validity from the Verification and Validation Square we adopt in Chapter 1. We argue theoretical performance validity, the generalization of the methods beyond their test problems, for the framework as a whole in Section 9.8.

9.8 Theoretical Performance Validity

In Chapter 1 we adopt the Verification and Validation Square, and in every research chapter we establish three of its four quadrants, reading theoretical structural validity from the soundness of the constructs, empirical structural validity from the suitability of the test problem, and empirical performance validity from the results. We defer the fourth quadrant, theoretical performance validity, to this chapter in each of those chapters. We take it up here.

Through theoretical performance validity we address the usefulness of the framework beyond the six test problems and six domains on which we develop it. We establish it in three parts. We first establish that each test problem is representative of a general class of problems rather than an isolated case. We then establish that we hold the completeness of the framework across unrelated domains, so that it is not an artifact of the domains we choose. We then strengthen confidence in the methods by identifying, for each, the structure that is invariant across domains and the structure that is specific to the domain on which we demonstrate it.

9.8.1 The test problems as representatives of general classes

We characterize each test problem by the properties on which we place it in a class, so that a designer recognizes a new problem as a member of the same class and applies the corresponding method to it.

The artificial light at night and bird strikes problem of Chapter 3 is representative of the class of problems in which we trace an outcome to signals originating in more than one of the cyber, physical, and social spaces, in which we hold design-relevant information in the cross-space causal structure that we lose through a single-space representation, and in which a designer requires an interpretable attribution of the outcome to its contributing factors rather than prediction alone. We place a problem in this class when we trace its outcome jointly across spaces and we value its interpretation as much as its accuracy.

The Agriculture 5.0 problem of Chapter 4 is representative of the class of problems in which we distribute decisions across hierarchical levels, in which the levels are conditioned on one another so that we alter the conditions for decisions at one level through the decisions we take at another, in which we find the couplings among decisions bidirectional rather than sequential, and

in which a designer requires a solution we support across a range of stakeholder priorities rather than at a single assumed configuration. We place a problem in this class when its decisions are multi-level, coupled, and subject to priority uncertainty.

The United Arab Emirates sustainable development problem of Chapter 5 is representative of the class of problems in which a designer works toward a specified future state across a horizon long enough that we encounter changed conditions along the way, in which we derive a strategy sound at the current state from the target rather than by projection from the present, in which we require robustness at every transition rather than at the endpoint alone, and in which we revise the goals on the performance we observe along the way. We place a problem in this class when it is goal-directed, multi-state, and long enough that we encounter changed conditions within it.

The hot rod rolling problem of Chapter 6 is representative of the class of problems in which the number of goals, the dimensionality of the decision space, and the density of trade-offs lie beyond what a designer navigates through conventional low-dimensional visualization, in which we hold design knowledge in structured scenario data but not in a form a designer queries directly, and in which a designer requires interpretable guidance and a recovery where a stated intent proves infeasible. We place a problem in this class when we cannot navigate its satisficing space unaided and we hold its knowledge latent in scenario data.

The supply chain resilience problem of Chapter 7 is representative of the class of problems in which we encounter changed operating conditions during operation rather than between design cycles, in which we read that change as signal rather than as noise, in which the standard for acceptable performance at the start of an operating period is no longer the right standard by its end, and in which a designer requires decisions at the timescale of the system under guarantees a high-stakes setting cannot do without. We place a problem in this class when we take non-stationarity as a defining condition and the decision interval is shorter than a human design cycle.

The carbon emission taxation problem of Chapter 8 is representative of the class of problems in which the success of a technically sound intervention rests on how the stakeholders we reach respond, in which they respond affectively, through memory, and through social transmission rather than as we would predict through a rational-actor model, and in which a designer requires diagnostic information about resistance before deployment rather than after. We

place a problem in this class when we trace outcomes to social legitimacy and stakeholders respond through perception, emotion, and social influence.

9.8.2 The completeness of the framework across unrelated domains

We demonstrate the six methods on six domains that share no subject matter. Aviation and lighting ecology, agricultural policy, national sustainable development, steel manufacturing, supply chain operation, and carbon taxation have no common vocabulary, no shared variables, and no overlapping stakeholders. In Section 9.6 we establish that the five layers operationalize all eleven characteristics of Table 2.3, and we establish it while drawing the demonstration of each characteristic from these unrelated domains.

We rest the argument on how the evidence is distributed. If the eleven-requirement completeness of the framework were an artifact of a domain, we would expect the coverage to concentrate in the domain whose structure happened to suit the requirements list, and to thin where the domain and the requirements were less well matched. We find no such concentration. Scenario Recognition is demonstrated across all six domains, Robustness and Context Awareness across three each, and Decentralized Decision Making, Self-Adaptability, Resilience, and Modularity each across two. No single domain carries all eleven, and of the characteristics we operationalize in the supply chain domain, every one except Organizability we also operationalize in at least one unrelated domain. Where we operationalize a characteristic in aviation, in manufacturing, and in supply chain operation, we operationalize it through structure that we draw from none of those three domains on its own, since they hold nothing in common. We therefore rest the completeness on the framework rather than on any domain, which is the demonstration we identify in Chapter 1 as the second element of theoretical performance validity, that the completeness of the framework is not an artifact of the domains chosen for validation.

9.8.3 The invariant structure of each method

We strengthen confidence in each method by separating what is specific to the domain of its test problem from what is invariant across the class. We take the invariant structure as what a designer carries to a new domain.

In the observation method of Chapter 3, the domain-specific elements are the particular variables of aviation, lighting, and migration and the particular attribution values we obtain. We hold as invariant the assignment of variables to the cyber, physical, and social spaces before model construction and the attribution of importance back to those spaces afterward. A designer applies the method to any problem in the class by assigning that problem's variables to the three spaces and attributing its outcome to them.

In the decision structuring method of Chapter 4, the domain-specific elements are the agricultural goals, variables, and the particular couplings we find. We hold as invariant the formulation of coupled multi-level decisions as compromise Decision Support Problems with robustness enforced through Design Capability Index constraints. A designer applies the method to any problem in the class by encoding that problem's decision types, levels, and couplings and enforcing DCI robustness across its priority range.

In the adaptive design method of Chapter 5, the domain-specific elements are the UAE's temporal states, goals, and the particular goal revisions we make. We hold as invariant the inverse derivation from a target state, the enforcement of DCI constraints at every transition, and the feedback goal-updating rule. A designer applies the method to any problem in the class by specifying that problem's target state and working backward under transition-level robustness with feedback revision.

In the augmentation method of Chapter 6, the domain-specific elements are the steel manufacturing scenarios, objectives, and the particular heuristics we extract. We hold as invariant the three-stage sequence of knowledge creation, knowledge exploration, and adaptable solution-space generation, in which we validate quantitatively before ingestion into the Knowledge Graph. A designer applies the method to any problem in the class by extracting and validating that problem's heuristics into a graph and querying it under the same three stages.

In the execution method of Chapter 7, the domain-specific elements are the inventory states, the supply chain facts, and the particular aspirations we compile. We hold as invariant the Compiled-Aspiration MDP, the compiler operator under its axioms, the CA-RS-Q algorithm with its separated channels, and the four guarantees, none of which rests on the supply chain domain. We prove the theorems for any admissible compiler and any knowledge graph, and we thereby place the invariance on a formal footing rather than an empirical one. A designer applies the

method to any problem in the class by supplying that problem's knowledge graph and process to the same construction.

In the assessment method of Chapter 8, the domain-specific elements are the carbon policy scenarios, the seven stakeholder classes, and the particular coalitions that form. We hold as invariant the emotion and memory substrate grounded in the circumplex model, arousal as a proxy for epistemic uncertainty, and the neuro-symbolic grounding of every action against a shared auditable world. We commit the architecture to no feature of the carbon policy domain, and we state its domain generality directly in Chapter 8. A designer applies the architecture to any problem in the class by instantiating that problem's stakeholder classes on the same cognitive substrate.

9.8.4 The scope and the limit of the generalization argument

We establish that each test problem represents a general class, that we hold the completeness of the framework across unrelated domains, and that we carry from each method a structure invariant across its class. For the execution method of Chapter 7, we rest the invariance on formal results that we prove independently of the domain, and we take this as the strongest form of the argument. For the other five methods, we rest the invariance on the separation of domain-specific inputs from domain-independent structure, supported by the demonstration on the test problem.

We do not establish the generalization through application to domains outside this dissertation. A designer strengthens theoretical performance validity most directly by applying a method to a problem in the same class in a domain we do not study here, and observing that we carry the invariant structure as we predict. We identify that application as the most direct strengthening available to us, and we take it up as future work in Section 9.12. What we establish here is that we structure the framework for that application, through classes a designer recognizes and methods whose invariant content we separate from the domains on which we demonstrate them. We record the argument in Table 9.7. With the argument of this section in place, we close the fourth quadrant of the Verification and Validation Square we open in Chapter 1, and we complete the validation strategy we adopt there, as we show in Figure 9.2.

Table 9.7: Problem Class Characteristics and Invariant Method Structure

Chapter and method	**Test problem**	**Defining characteristics of the class**	**Domain-specific elements**	**Invariant structure carried to a new domain**
Ch 3, CPSS-structured predictive modeling	ALAN and bird strikes	Outcome traced jointly across spaces; cross-space causal structure design-relevant; interpretation as valuable as accuracy	Aviation, lighting, migration variables and their attribution values	Assignment of variables to the three spaces before modeling; attribution back to those spaces after
Ch 4, Integrated Policy Designing Decision Model	Agriculture 5.0	Multi-level decisions; levels conditioned on one another; couplings bidirectional; robustness required across priorities	Agricultural goals, variables, and the couplings found	Coupled multi-level cDSP formulation with DCI robustness enforced
Ch 5, Goal-Oriented Inverse Design Method	UAE sustainable development	Goal-directed across a long horizon; conditions changed along the way; robustness required at every transition; goals revisable on feedback	UAE temporal states, goals, and goal revisions	Inverse derivation from target; DCI at every transition; feedback goal updating
Ch 6, GenAI and	Hot rod rolling	Satisficing space beyond unaided	Steel manufacturing	Three-stage creation,

Knowledge Graph framework		navigation; knowledge latent in scenario data; interpretable guidance and infeasibility recovery required	scenarios, objectives, and extracted heuristics	exploration, generation with validation before ingestion
Ch 7, Compiled-Aspiration MDP and CA-RS-Q	Supply chain resilience	Non-stationarity a defining condition; change read as signal not noise; decision interval shorter than a design cycle; guarantees required	Inventory states, supply chain facts, compiled aspirations	Compiled-Aspiration MDP, compiler axioms, separated channels, four guarantees; invariance on a formal footing
Ch 8, Cognitively grounded stakeholder agents	Carbon taxation	Success traced to stakeholder response; response affective, memory-laden, socially transmitted; diagnosis required before deployment	Carbon policy scenarios, seven stakeholder classes, coalitions formed	Emotion-memory substrate, arousal as uncertainty proxy, neuro-symbolic grounding

We establish that each test problem represents a general class and that each method carries structure invariant across that class. For the method of Chapter 7 the invariance rests on formal results that hold independently of the domain. For the other five methods the invariance rests on the separation of domain-specific inputs from domain-independent structure, supported by the demonstration on the test problem. We take up application to domains outside this dissertation as future work in Section 9.12.

9.9 Answering the Principal Research Question

In Chapter 1 we state the Principal Research Question, its hypothesis, and its expected new knowledge, and we identify Chapter 9 as the place where we answer it. We answer it here, through the assembled framework of the preceding sections.

We state the Principal Research Question in Chapter 1 as follows.

> *Principal Research Question (PRQ). What are the principles and methods that designers can use to strategically engineer robust, adaptive, and intelligent interventions in Evolving Cyber-Physical-Social Systems across state transitions under uncertainty?*

We state the Hypothesis for the Principal Research Question as follows.

> *Hypothesis (PRQ). Through the progressive development and integration of principled methods for state observation, robust and adaptive decision structuring, augmented intelligence, governed autonomous execution, and social legitimacy assessment, designers can strategically engineer interventions in Evolving Cyber-Physical-Social Systems that are robust under uncertainty, adaptive across state transitions, and socially legitimate before deployment.*

We answer through the Decision-State Information Model. The principles are the two commitments we carry from Chapter 1 through every layer, that a designer resolves the information asymmetry and the intelligence asymmetry of an E-CPSS together rather than separately, and that a designer authorizes an advance of the system only when the strategy is robust, executable, and socially legitimate at once. The methods are the six we develop across Chapters 3 through 8, assembled into the five layers and the transition condition of the DSIM. Through the framework, a designer reads the state of an E-CPSS, structures decisions within it, derives the pathway across its transitions, navigates trade-offs where they cannot navigate the problem unaided, executes where they cannot match the pace of the system through the design cycle, assesses social legitimacy before deployment, and authorizes an advance only under the conjunction of robustness, knowledge retention, aspiration satisfaction, and social legitimacy.

We map the terms of the Principal Research Question onto the framework directly. A designer engineers interventions that are robust under uncertainty through the Design Capability Index constraints of Layer 2, enforced within a state in Chapter 4 and at every transition in Chapter 5. A designer engineers interventions that are adaptive across state transitions through the inverse design of Layer 2 and the autonomous execution of Layer 4, through which we adapt the strategy as we carry the system forward and as we encounter changed conditions during operation. A designer engineers interventions that are intelligent through the augmentation of Layer 3, which extends cognitive reach without displacing judgment, and through the observation of Layer 1, which grounds every subsequent decision in a design-relevant characterization of the state. A designer engineers interventions that are socially legitimate before deployment through the assessment of Layer 5, which produces the stakeholder response before the strategy is committed. We carry the word across in the question through the transition condition and the learning loop, which are the elements of the framework we exercise between states rather than within one. We draw the evidence for the answer from the preceding sections, from the operationalization of the eleven requirements in Section 9.6, from the consistency of every hypothesis with its test problem in Section 9.7, and from the theoretical performance validity of Section 9.8.

The new knowledge we produce we state as follows.

> *New Knowledge (PRQ). The Decision-State Information Model: a state-indexed design intelligence framework for the strategic engineering of Evolving Cyber-Physical-Social Systems.*

9.9.1 The scope of the answer

We state the scope of the answer plainly, so that a reader does not take what we establish for more than it is. We arrive at the DSIM as a retrospective synthesis of six methods we develop against six challenges, rather than as a prospective template we specify in advance and then fill in. We assemble the framework and argue its coherence; we do not present it as a recipe a designer follows from the top.

We develop and demonstrate the six methods each on its own test problem, in its own domain, and we assemble them into the DSIM in this chapter. We do not execute the five layers end to end on a single E-CPSS, carrying one system from observation through assessment and

across a transition within one continuous application. We assemble the framework and argue its completeness, and we have yet to execute all five layers on one system. We identify this as the primary scope boundary of the dissertation, and we take up integrated deployment as the first direction of future work in Section 9.12.

We are equally clear about what we do not concede through this boundary. We demonstrate each layer on a test problem we choose to exercise it, we establish the composition relationships among the layers in Section 9.5 and rest them on dependencies we carry across the chapters, and we assemble the transition condition from criteria each of which we demonstrate in the chapter from which we obtain it. What remains is the single continuous application, not the validity of the layers or of their composition. We hold the answer to the Principal Research Question to that scope, the framework assembled and argued rather than executed end to end, and we take the integrated application as the confirmation we obtain at the next stage of this research.

9.10 Contributions and New Knowledge

We organize the contributions of this dissertation in two tiers, following Boyer's taxonomy as we adopt it in Chapter 1. At the level of method, we contribute through the Scholarship of Discovery, across six research chapters in each of which we develop new methods, mathematics, or architectures. At the level of synthesis, we contribute through the Scholarship of Integration, assembling the six into the Decision-State Information Model. We state each contribution as what we develop, what we demonstrate, and what new knowledge we produce, and we reproduce each new knowledge statement as we state it in Chapter 1. We record the contributions in Table 9.8.

Boyer's enlargement of scholarship. In *Scholarship Reconsidered*, Boyer argues that the academy had narrowed the meaning of scholarship until it stood for basic research alone, with every other academic function treated as something that flows from research rather than as scholarship in its own right. He objects that knowledge does not develop in that linear a manner, and that the arrow of causality runs in both directions. In place of the narrowed definition he proposes four separate and overlapping functions, the scholarship of discovery, of integration, of application, and of teaching. Two of these bear on the work of this dissertation, and we take them up in turn.

Discovery and integration. Discovery is the function that comes closest to what the academy ordinarily means by research. Boyer describes it as the disciplined pursuit of an investigation wherever it leads, held to a commitment to knowledge for its own sake. Integration he defines as the work of giving meaning to isolated facts by placing them in perspective, making connections across disciplines, setting a specialty in a larger context, and interpreting findings so that a wider reading of them becomes available. He is careful to distinguish this from dilettantism, and insists on serious and disciplined work that draws findings together and brings new insight to bear on original research. Boyer describes integration as working in two ways: at the boundaries where fields converge, in what he calls overlapping neighborhoods, and through interpretation, by fitting findings into a larger intellectual pattern. He holds the second to be increasingly necessary, since specialization without a broader perspective risks pedantry, and he argues that it is through connectedness that research is finally made authentic. Boyer draws the line between the two functions through the questions they pose. A scholar engaged in discovery asks what is to be known and what is yet to be found. A scholar engaged in integration asks what the findings mean, and whether it is possible to interpret what has been discovered so as to yield a larger and more comprehensive understanding.

We follow this taxonomy in this dissertation. In Chapters 3 through 8 we ask Boyer's first question. At each of the six escalating design challenges we ask what a designer of an Evolving Cyber-Physical-Social System needs and does not yet have, and we answer by developing something that did not exist beforehand. Through Chapter 3 we develop a method, through Chapters 4 and 5 a method and its mathematics, through Chapter 6 a framework, through Chapter 7 a formal construction and the theorems that support it, and through Chapter 8 an architecture. We demonstrate each on a test problem drawn from a real design situation, we establish its validity through the Verification and Validation Square, and we state precisely what it adds to what was known, which is the new knowledge we record for each research question in Section 9.7. Six questions, six methods, six statements of new knowledge. This is our contribution through the scholarship of discovery. The six methods are new. The principles and the mathematics on which they rest we create in part in this work and adopt in part from established theory, and in each research chapter we mark which is which, so that a reader sees precisely where the discovery lies. The compromise Decision Support Problem, the Design Capability Index, the circumplex model of affect, and the Markov decision process we adopt; the Integrated Policy Designing Decision

Model, the Goal-Oriented Inverse Design Method, the compiler that sets an aspiration from a knowledge graph, and the memory-emotion substrate we build. Discovery lies in what we build on the foundation we adopt.

In this chapter we ask Boyer's second question. We ask what those six discoveries mean when we hold them together, and we answer that they compose the Decision-State Information Model. The answer rests on connections we establish rather than assert. At each of the five layers we close an asymmetry at an order of complexity we reach at no other layer, as we show in Section 9.4. Where the layers meet, functionalities arise that we hold at no single layer, and each rests on a construct we carry forward from one chapter into another, as we show in Section 9.5. Through the assembled framework we operationalize all eleven design requirements of Chapter 2, which we do through no method alone, as we show in Section 9.6. And the completeness holds across six unrelated domains rather than resting on any one of them, as we show in Section 9.8. What the six discoveries hold together is a structure none of them holds separately, and that structure is an architecture for the strategic engineering of Evolving Cyber-Physical-Social Systems. This is our contribution through the scholarship of integration, and it is the connectedness Boyer asks integration to produce.

9.10.1 The Method-Level contributions

Contribution 1, State Observation. We develop a CPSS-structured predictive modeling method in which we assign observational variables to the cyber, physical, and social spaces before model construction and attribute importance back to those spaces afterward. We demonstrate on the artificial light at night and bird strikes test problem that we obtain through the method a state characterization interpretable as a description of the tri-space state while remaining comparably accurate, and that light pollution level is the strongest predictor of bird strike incidence among the factors we consider. We construct I(si), the first component of the DSIM.

> *New Knowledge 1. A CPSS-structured predictive modeling method for constructing interpretable tri-space state characterizations of Evolving Cyber-Physical-Social Systems to support design decisions. Empirical confirmation that Artificial Light at Night is a statistically significant contributing factor to bird strikes, with light*

pollution level identified as the strongest predictor, with implications for public policy and aviation safety.

Contribution 2, Robust Decision Structuring. We develop the Integrated Policy Designing Decision Model, a coupled compromise Decision Support Problem formulation that encodes decision types, interaction strengths, and hierarchical levels across macro, meso, and micro scales, with Design Capability Index constraints enforcing robustness. We demonstrate on the Agriculture 5.0 test problem that three of six decision variables interact bidirectionally and are better formulated as coupled than as sequential, and that the satisficing solution space holds across the stakeholder priority scenarios we examine. We construct D(si), the intra-state decision map.

New Knowledge 2a. A coupled decision method anchored in the Integrated Policy Designing Decision Model and Design Capability Indices for designing robust multi-level satisficing strategies in Evolving Cyber-Physical-Social Systems under uncertainty.

Contribution 3, Adaptive Strategy Design. We develop the Goal-Oriented Inverse Design Method, through which we work backward from a desired future state, enforce Design Capability Index constraints at every transition, and updates goals through feedback when observed performance departs from projection. We demonstrate on the United Arab Emirates sustainable development test problem across three temporal states that we obtain through the method adaptive strategies that remain robust across the transitions we model. We construct D(si to si+1), the inter-state decision map, and with it we complete the Decision Layer.

New Knowledge 2b. A Goal-Oriented Inverse Design Method anchored in the principle of working backward from desired future states, with DCI-enforced transition conditions and feedback-driven goal updating, for deriving adaptive strategies across E-CPSS state transitions under uncertainty.

Contribution 4, Augmented Design Intelligence. We develop a generative AI framework in which a large language model extracts knowledge statements from structured scenario data, we validate each statement quantitatively before admission to the Knowledge Graph, and we support interactive querying and adaptable solution-space generation through the validated graph. We demonstrate on the hot rod rolling steel manufacturing test problem that a designer navigates a high-dimensional satisficing space with interpretable guidance and recovers an adaptable solution

space when a stated intent proves infeasible. We construct Layer 3, the Augmentation Layer, and we establish the Knowledge Graph on which we draw in the two subsequent layers.

> *New Knowledge 3a. A GenAI-anchored Knowledge Graph framework for augmenting E-CPSS design decision-making, with LLM-based knowledge extraction, quantitative validation protocols, and interactive trade-off summarization for interpretable guidance in high-dimensional design spaces.*

Contribution 5, Autonomous Strategy Execution. We develop the Compiled-Aspiration MDP and the CA-RS-Q algorithm, in which we compile the agent's aspiration from a dynamically updated Knowledge Graph and hold the value-learning channel apart from the aspiration channel, and we prove convergence under a settled graph, a finite-time satisficing rate under a stationary graph, and robustness of policy value under graph perturbation. We demonstrate on the supply chain resilience test problem that we enable the agent to follow a structural change and to recover once it is withdrawn, with each decision auditable to the facts from which we set the aspiration. We construct Layer 4, the Execution Layer, contributing to M(si) and K(si).

> *New Knowledge 3b: A method for sourcing a reinforcement-learning agent's satisficing aspiration from a structured, dynamically updated Knowledge Graph, the Compiled-Aspiration MDP and the CA-RS-Q algorithm which keeps value learning and aspiration formation in separate channels, carries convergence, finite-time satisficing-rate, and graph-perturbation robustness guarantees, and renders each decision auditable to the facts that set the aspiration, for autonomous strategic execution in non-stationary E-CPSS.*

Contribution 6, Social Legitimacy Assessment. We develop a cognitively grounded stakeholder agent architecture in which we govern the internal state of each agent through a deterministic emotion and memory substrate, we take arousal as a proxy for epistemic uncertainty through which we scale the strength of the bias of emotional state on memory retrieval, and we ground every action against a shared auditable world through a neuro-symbolic layer. We demonstrate on the carbon emission taxation test problem that we obtain behavior measurably distinct from rational-actor prediction, that we find structured memory the most consequential mechanism, and that we measure a distinct departure from rationality for each stakeholder class. We construct Layer 5, the

Social Assessment Layer, through which we produce the social legitimacy assessment from which we obtain the fourth criterion of the transition condition.

> *New Knowledge 4. A cognitively grounded stakeholder agent architecture with predictive-processing-anchored memory-emotion coupling, neurosymbolic knowledge graph grounding, and automated agent generation for pre-deployment social legitimacy assessment of adaptive interventions in Evolving Cyber-Physical-Social Systems.*

9.10.2 The synthesis contribution

The synthesis contribution is the Decision-State Information Model. We assemble the six method-level contributions into the six-tuple {S, I, D, M, K, T}, in which we construct the five operational layers through the six methods, we feed the knowledge we produce at each state forward into the observation of the next through the connection of K(si) into I(si+1), and we authorize an advance through the transition condition only under the conjunction of robustness, knowledge retention, aspiration satisfaction, and social legitimacy. In Section 9.5 we establish that we hold functionalities across the assembly that we hold at no single method, and in Section 9.6 we establish that we operationalize all eleven E-CPSS design requirements of Chapter 2 through the five layers across six unrelated domains. This is the contribution at the level of the Scholarship of Integration, that we obtain from the six discoveries together a structure we obtain from none of them alone, and that we hold that structure as an architecture for the strategic engineering of Evolving Cyber-Physical-Social Systems.

> *New Knowledge (PRQ). The Decision-State Information Model: a state-indexed design intelligence framework for the strategic engineering of Evolving Cyber-Physical-Social Systems.*

9.10.3 What a designer can do through the framework

We close the account of the contributions by stating what a designer does through the framework. Through the DSIM, a designer reads the state of an E-CPSS across its three spaces in a form that supports design rather than description alone. A designer structures the decisions warranted within a state, with their couplings explicit and their robustness enforced, and derives the pathway across

a transition toward a specified future state. A designer navigates a satisficing space too large for unaided navigation, and executes strategy at the timescale of a system that moves faster than the design cycle, under guarantees that a high-stakes setting requires. A designer produces the social response to a strategy before committing to it, and carries the knowledge produced at each state forward into the characterization of the next. And through the transition condition, a designer determines whether the system may advance, under a standard that holds robustness, knowledge retention, aspiration satisfaction, and social legitimacy together rather than trading one against another. These are the capabilities we provide through the framework, and together we take them as the design intelligence the strategic engineering of an E-CPSS requires.

Table 9.8: Contributions and New Knowledge of the Dissertation

Contribution	Chapter and RQ	What we develop	What we demonstrate	DSIM component	New knowledge
1. State observation	Ch 3, RQ1	CPSS-structured predictive modeling with SHAP tri-space attribution	Interpretable tri-space characterization; ALAN a significant contributor to bird strikes	I(si)	NK1
2. Robust decision structuring	Ch 4, RQ2a	Integrated Policy Designing Decision Model, coupled cDSP with DCI	Three of six variables coupled; satisficing region holds across priorities	D(si)	NK2a
3. Adaptive strategy design	Ch 5, RQ2b	Goal-Oriented Inverse Design Method	Adaptive strategies robust across three temporal states	D(si to si+1)	NK2b
4. Augmented design intelligence	Ch 6, RQ3a	GenAI and Knowledge Graph framework with validation before ingestion	Interpretable navigation and infeasibility recovery in a high-dimensional space	Layer 3, Knowledge Graph	NK3a
5. Autonomous execution	Ch 7, RQ3b	Compiled-Aspiration MDP, CA-RS-Q, four theorems	Followed and recovered under structural change; convergence, satisficing rate, graceful degradation, auditability	M(si), K(si)	NK3b

6. Social legitimacy assessment	Ch 8, RQ4	Cognitively grounded stakeholder agent architecture	Behavior measurably distinct from rational-actor; memory most consequential	Layer 5, legitimacy assessment to T	NK4
Synthesis contribution	**Ch 9, PRQ**	**Assembly of the six into the DSIM, {S, I, D, M, K, T}**	**Functionalities we hold at no single method; eleven requirements operationalized across six unrelated domains**	**The full tuple, T assembled**	**NK (PRQ)**

We reproduce each new knowledge statement in full in the text of Section 9.10. We make the six method-level contributions through the Scholarship of Discovery, and the synthesis contribution through the Scholarship of Integration.

9.11 Limitations and Opportunities for Improvement

We state the limitations of the dissertation at three levels. We first state the scope limitation of the framework as a whole. We then state the method-level limitations, which we frame as the assumptions we make in each method and the conditions under which those assumptions bind. We then state the framework-level limitations, which we trace to a single design commitment we carry through the DSIM. We carry each limitation into Section 9.12, where we grow the directions of future work from these edges.

9.11.1 The Scope limitation

We assemble the framework and argue for its completeness, but we do not execute all five layers from beginning to end on a single E-CPSS. We develop and demonstrate each layer on a test problem chosen to exercise it, we establish the composition relationships among the layers in Section 9.5, and we do not carry one system continuously from observation through assessment and across a transition within one application. This is the primary scope boundary of the dissertation. We rest the composition relationships on dependencies we carry across the chapters, and we take their confirmation in a single continuous run as the strongest strengthening of the synthesis claim available to us. We take this up as the first direction of Section 9.12.

A second element of scope concerns the reach across domains. We demonstrate the six methods on six unrelated domains, which establishes that the completeness of the framework is not an artifact of any one of them, and which is distinct from demonstrating that we carry a method to a domain we do not study. Through the invariance argument of Section 9.8 we establish that we structure the methods to travel, and we identify the confirmation that they travel to an untested domain as an empirical step in Section 9.12.

9.11.2 Method-Level limitations

We trace several of the method-level limitations to one root. Through the DSIM we keep the designer in the loop as the source of the goals, the priorities, and the standards against which we operate the framework, which is a deliberate commitment of the approach rather than an oversight, and which is also where we find the sharpest boundaries of the individual methods. We state each

limitation as the assumption we make in the method and the condition under which that assumption binds.

Contribution 1, State Observation. Through the method we establish the statistical contribution of each feature to the outcome, under the assumption that we observe a feature set containing the operative factors. Where we trace the outcome to an unobserved factor, or where a designer requires the causal mechanism rather than the association, we take the attribution we obtain through the method as a starting point for causal investigation rather than as its conclusion. We identify the opportunity in Section 9.12 as a state-observation method through which we carry from statistical contribution toward causal mechanism.

Contribution 2, Robust Decision Structuring. Through the method we enforce robustness across the stakeholder priority configurations a designer specifies, under the assumption that we specify configurations representing the priorities we will encounter. Where we encounter realized priorities outside the specified range, we hold the robustness we establish as relative to that range. We identify the opportunity as a principled construction of the priority range itself, so that we establish robustness against the priorities we may encounter rather than against the priorities a designer anticipates.

Contribution 3, Adaptive Strategy Design. Through the method we derive a robust pathway to a specified target state, under the assumption that we choose the target state well. Through the method we revise goals on feedback where observed performance departs from projection, and we do not evaluate whether the target itself is the right destination. Where a designer specifies a target we cannot or should not reach, we obtain through the method a well-formed pathway to a poorly chosen end. We identify the opportunity as a coupling of target selection to the observation and assessment layers, so that we test the destination as we derive the pathway to it.

Contribution 4, Augmented Design Intelligence. Through the framework we admit heuristics we validate quantitatively against the scenario data from which we draw them, under the assumption that we assemble a scenario set representing the conditions in which we will apply those heuristics. We carry a heuristic supported by the scenarios we examine to conditions outside the set only insofar as we assemble that set representatively. We demonstrate adaptivity as interactive exploration of an existing solution space, and we identify the opportunity as its extension to the generation of new solutions through fresh evaluation.

Contribution 5, Autonomous Strategy Execution. We prove the guarantees in a finite tabular setting, under the assumption that we represent the state space tabularly. For a large state space we require function approximation, and we would need to re-establish the convergence, satisficing-rate, and robustness results in that setting, which we identify in Chapter 7 as a further step. In stationary and well-specified regimes we hold the agent deliberately conservative, and through the method we trade raw return for the guarantees a non-stationary high-stakes setting requires. We identify the opportunity as the extension of the guarantees to function approximation, through which we carry the method from the benchmark scale toward the scale of a deployed system.

Contribution 6, Social Legitimacy Assessment. Through the architecture we ground the cognitive parameters of the stakeholder agents in theory, in the circumplex model of affect and a predictive-processing account of arousal, under the assumption that we obtain diagnostically informative behavior from theoretically grounded parameters. We do not calibrate the parameters against observed behavioral data from the stakeholders a real intervention would reach, and we obtain through the architecture a structured account of where we expect resistance to form rather than a calibrated forecast. We identify the opportunity as the calibration of the cognitive substrate against behavioral data, through which we carry the assessment from diagnostic information toward validated prediction. Through this contribution we also incur a computational cost that constrains the scale at which we apply it, and we analyze that cost in Chapter 8. We issue a language-model call for every agent decision, so we make a number of calls that scales with the product of the agent count, the horizon, and the deliberation rounds per turn, and we therefore pay a cost that grows multiplicatively rather than linearly as we enlarge the problem. We compound this cost through the determinism on which we rest the auditability of the method, since we execute every call serially at a fixed temperature to hold a run reproducible from its seed, and in doing so we forgo the parallel execution the hardware is built for. We regard this cost as the present price we pay for a reproducible and auditable simulation rather than as an incidental inefficiency, and we identify the reduction of that cost without the loss of reproducibility as a distinct opportunity, which we pursue through caching and reuse of agent reasoning, through distillation into smaller models, and through forms of parallel execution we engineer to remain numerically deterministic.

9.11.3 Framework-level limitations

We trace the framework-level limitations to one open problem, which is our reliance on inputs the designer supplies. In the DSIM a designer specifies the set of states, sets the thresholds inside the transition condition, and chooses the target state from which the adaptive pathway is derived. This reliance is the same design commitment from which we obtain the method-level boundaries of Section 9.11.2, seen now at the level of the framework, and through it we open three specific problems.

The states are specified rather than detected. A designer specifies the set of states S, and we do not detect through the framework the moment at which one state becomes the next. In every test problem we take the states from the structure of the design situation, from a planning horizon, an operating cycle, or a policy announcement. A framework a designer relies on to know that the system has transitioned would hold a capacity for state detection that we do not currently provide through the DSIM. We identify the opportunity as a detection mechanism through which we read a transition from the observation layer rather than receive it from the designer.

The thresholds in the transition condition are set rather than derived. We establish the structure of the transition condition and the necessity of its conjunctive form, and a designer sets the threshold inside each of its four criteria. We do not derive through the framework where the robustness threshold, the retention standard, the aspiration target, or the legitimacy standard should sit, and we take each from the requirements of the design situation. We identify the opportunity as a principled derivation of these thresholds, so that we ground the condition governing a state advance in the system rather than in the designer's judgment alone.

The composition is established rather than exercised in one application. We establish the functionalities of Section 9.5 that we hold where layers meet through the dependencies among the chapters, and we do not exercise them together in one continuous run. This is the framework-level counterpart of the scope limitation of Section 9.11.1, seen from the side of the composition. We identify the opportunity as the integrated deployment we set out in Section 9.12 as the first direction of future research.

9.12 A Vision for Future Research

The Decision-State Information Model is a platform, and we pursue through it a program of research that reaches past the completion of the present work. We set out our vision in two groups. Through the first group we consolidate the extensions that strengthen and complete the framework, and we state these compactly. Through the second group we identify research avenues that we open through the dissertation and that stand as questions in their own right, in decision theory, in reinforcement learning, in the study of bounded rationality, in machine cognition, and in the design of governable autonomy.

9.12.1 Consolidating and completing the framework

Direction 1, Integrated Deployment and Cross-Domain Confirmation. We carry one E-CPSS continuously through all five layers and across a transition, exercising the composition relationships of Section 9.5 together in one run, and we apply each method to a problem in the same class in a domain we do not study here, confirming that the invariant structure of Section 9.8 carries as the argument predicts. Through the first move we confirm the assembly, and through the second we convert the theoretical performance validity of Section 9.8 from a structural argument into an empirical one. Through both together we answer the scope and domain-reach limitations of Section 9.11.1.

Direction 2, Deepening the Individual Layers. We deepen each layer along the line we identify through its limitation, carrying state observation from statistical contribution toward causal mechanism in Layer 1, constructing the priority range rather than receiving it and coupling target selection to observation and assessment in Layer 2, extending augmentation from the exploration of an existing solution space toward the generation of new solutions in Layer 3, extending the execution guarantees from the finite tabular setting to function approximation in Layer 4, and calibrating the cognitive substrate against behavioral data in Layer 5. Through these we answer the method-level opportunities of Section 9.11.2, and through each we deepen a layer without altering the architecture in which we hold them together.

Direction 3, Closing the Designer-in-the-Loop Inputs. We develop the framework's capacity to detect a state transition from the observation layer rather than receiving the state set from the designer, and to derive the thresholds inside the transition condition from the system rather than

taking them from the designer's judgment. Through these we answer the two faces of the designer-supplied-input problem of Section 9.11.3, and together through them we carry the framework from one a designer configures at every state toward one in which we read the conditions for advancing from the system itself.

9.12.2 Research avenues we open through the dissertation

We identify five avenues. In each we take a commitment we make in the dissertation and pursue it as a research question in a recognized field. For each, we describe what researchers in that field achieve through established approaches, in terms of the functionality they provide, and we state what remains open against that boundary rather than in isolation.

Avenue 1, Context as a Structured and Evolving Object in Decision Theory. In the execution layer we treat the standard for acceptable performance as something we compile from a structured description of the world and revise as that description changes. In established treatments in decision theory, context is represented as a state index over utility functions, or as a reference point formed from a decision-maker's expectations, and in each the scale on which we value outcomes is conditioned on the context. A question that remains open is whether we model context better, for an evolving system, as a dynamic and structured knowledge object that conditions the aspiration itself, the standard for good enough, rather than the utility scale or the reference point, and what a normative account of choice looks like when we hold the context governing a decision as a structured artifact revised endogenously. The Compiled-Aspiration construct is one concrete instance of such a context, and we regard the theory that would situate it as belonging to decision theory rather than to the design of E-CPSS alone.

Avenue 2, Contextual Reinforcement Learning with Structured Context that Sets the Aspiration. In the execution layer we place a knowledge graph inside the decision process as a formal state variable and compile the agent's aspiration from it. In established approaches to contextual reinforcement learning, context is represented as a latent vector or a low-dimensional hidden parameter over the reward function or the transition dynamics, and much of the work in that field addresses how such a context is inferred from history. Two questions remain open against that boundary. We ask first whether we represent context more usefully, not as a latent embedding but as an inspectable and editable structure that a designer reads and corrects. We ask second, and this is the question nearer to our construction, whether context conditions the agent's aspiration

rather than its reward or its dynamics, with the channel that forms the aspiration held apart from the channel that learns value. To our knowledge, a structured and editable context that conditions a satisficing aspiration, held separate from value learning, has not been jointly addressed in this field, and we offer the Compiled-Aspiration MDP as a first formalization from which we pursue that question.

Avenue 3, Satisficing Under Evolving Aspiration. We adopt Simon's satisficing as our decision orientation and, in the execution layer, make the aspiration a dynamic object that we derive and revise. In established computational treatments, the aspiration is held in one of two ways, either as a moving function of past performance, or as an external target a policy is required to meet, propagated through a world model supplied to the agent. A question that remains open is what a theory of satisficing looks like when we compile the aspiration from an explicit model of the world and revise it as we revise that model, when the aspiration is neither a trace of past performance nor an external instruction, and when we raise the separation of aspiration formation from value learning to a general design principle rather than a feature of one algorithm. We establish for the Compiled-Aspiration MDP that we hold the satisficing guarantee under perturbation of the structure from which we compile the aspiration, and we regard that property, which established treatments do not reach, as the natural center of this avenue.

Avenue 4, Emotion and Memory as Functional Components of Machine Cognition. In the assessment layer we ground stakeholder behavior in a cognitive substrate, treating arousal as a proxy through which we scale the strength of the bias of emotional state on memory retrieval, and we establish that we obtain coherent behavior, measurably distinct from rational-actor prediction, from this substrate rather than from the language model we run on top of it. In established work on affective agents, emotion is most often computed through the language model, by appraisal expressed in the prompt, or through a retrieval function scored for recency, importance, and relevance over a memory stream. In separate accounts in cognitive science, arousal is related to the precision of inference, and arousal is related to the priority with which memory is encoded and retrieved, and these accounts stand apart from one another. A question that remains open is whether we treat emotion and memory better as a deterministic substrate beneath an action-only language model, rather than as behavior prompted from within it, and whether arousal understood as a proxy for epistemic uncertainty, scaling the strength with which affect biases retrieval, is the mechanism that connects the account of arousal as precision to the account of arousal as a bias on memory. To

our knowledge these accounts have not been jointly modeled in this way, and we offer the stakeholder architecture as a first demonstration from which we study the connection. We regard the questions here as belonging to the design of cognitively grounded agents wherever we model human judgment, in negotiation, in human-agent teaming, and in the simulation of social response at scale.

Avenue 5, Structured Knowledge as a Governable World Model. Across three chapters we use the knowledge graph in a way that grows past a store of heuristics. In the augmentation layer it holds validated design knowledge, in the execution layer we hold it as a formal state variable from which we compile the aspiration of an agent, and in the assessment layer we ground a population of agents against it as a shared auditable world. In established world models, the world is held in one of two ways, either implicitly, in weights a designer cannot inspect, or explicitly, as a structure built for the agent's own planning, while in established methods for governing an agent, the agent is held to a symbolic specification that is fixed rather than editable. A question that remains open is whether we hold a single inspectable and editable knowledge graph at once as the world model an agent navigates and executes against, as the artifact a designer audits and corrects and holds the agent accountable to, and as the shared grounding for multi-agent social simulation, and how we bound an agent's dependence on such a model when the model is wrong, which is the robustness property we already establish in the execution layer. To our knowledge we do not yet have this threefold use of one governable structure in the literature, and we regard it as connecting the dissertation to the wider problem of building autonomous systems whose model of the world is open to inspection and correction rather than sealed inside a form no designer reads.

9.12.3 Toward Self-Designing Evolving Systems

We close the vision by setting out where the program leads. Figure 9.3 holds the argument.

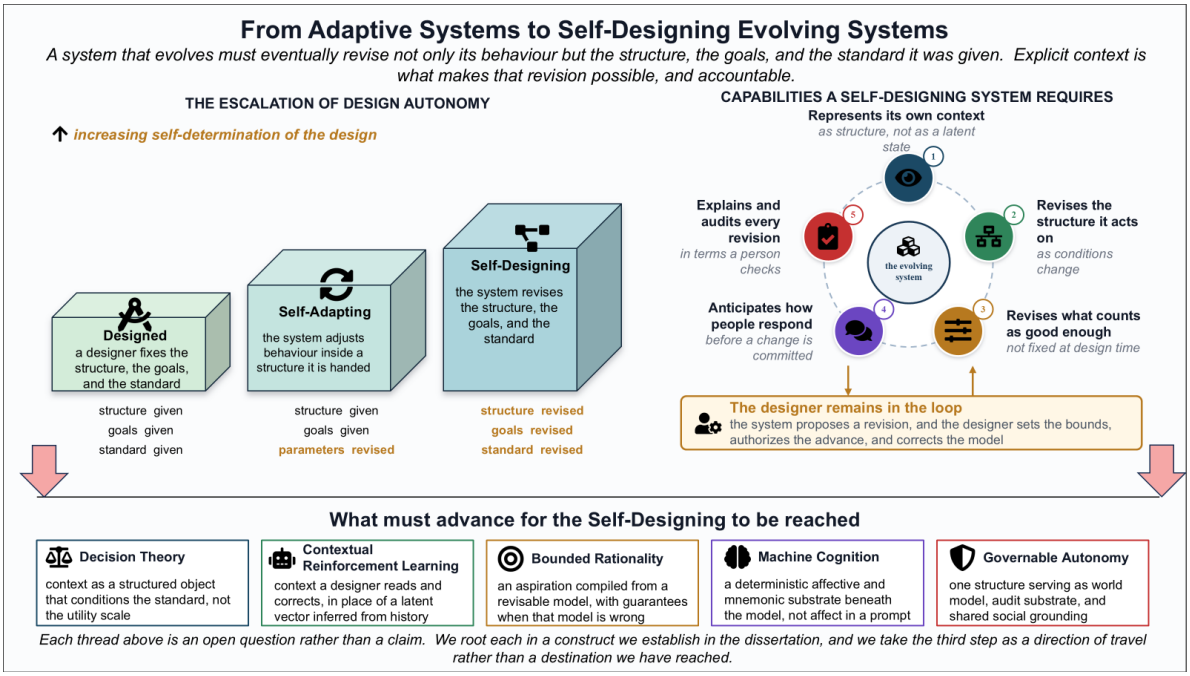


Figure 9.3: Towards Self Designing Systems

Three positions on the escalation. A designer stands in one of three places with respect to an evolving system, and we show these as the three ascending steps at the left of the figure. At the first, Designed, we fix the structure, set the goals, and state the standard by which we judge the result. At the second, Self-Adapting, the structure and the goals are still ours to fix, and the system adjusts its behavior inside the design we hand it. This is where adaptive and autonomic approaches sit. At the third, Self-Designing, the system revises the structure it acts on, the goals it pursues, and the standard by which it judges good enough.

In the ledger beneath the three steps we record what a designer gives and what the system revises at each position, across three rows: structure, goals, and standard. At the first position the system revises nothing. At the second it revises parameters alone. Only at the third does it revise all three.

Where the dissertation sits. We sit between the second step and the third. We are past the second because through the DSIM we do more than tune behavior inside a design handed to us. In Chapter 5 we revise the goals themselves through feedback when observed performance departs from projection, and in Chapter 7 we compile the standard for acceptable performance from a structured description of the world and revise it as that description changes, rather than fixing it at the moment

we formulate the problem. On the ledger, we move the second row and the third. We do not reach the third step because the first row stays fixed. A designer still supplies the structure, specifying the set of states, setting the thresholds inside the transition condition, and choosing the target from which the pathway is derived, as we set out in Section 9.11.3. What we offer, then, is the foundation for the third step rather than the step itself.

What the framework already supplies. Each of the five capabilities in the ring at the right of Figure 9.3 has some counterpart in the framework, and the gap between counterpart and capability is what remains. Layer 1 and the Knowledge Graph of Layer 3 hold context as structure rather than as a latent state, though a designer reads that structure and the system does not read it of itself. The Knowledge Graph of Layers 3 and 4 is editable and does change as facts arrive, though a designer performs the editing. Through the goal updating of Chapter 5 and the compiled aspiration of Chapter 7 we revise the standard rather than fix it, and we do so within bounds a designer sets. Through Layer 5 we anticipate how the people an intervention reaches will respond before a designer commits it, and of the five this is the capability we supply most completely. Through the per-decision auditability of Chapter 7 we trace every action to the facts that set its aspiration, which is explanation at the level of a decision rather than at the level of a revision to the design itself. Through the framework we reach the third step along some rows and not others, and that uneven pattern is the map of the work ahead.

What a self-designing system still needs. Three things are missing. We name the first two among the opportunities of Section 9.11.3, and the third follows from placing explanation and audit among the capabilities rather than beside them. The system needs to read a state transition for itself rather than receive the set of states from a designer. It needs a principled basis for the thresholds that govern an advance, rather than taking them from a designer's judgment. And it needs to carry explanation from the level of a single decision to the level of a change in its own design, so that when it proposes a revision it can say why. The first two we take up as the third consolidating direction of Section 9.12.1; the third belongs with the governable world model of Avenue 5. The remaining capabilities depend on the avenues of Section 9.12.2 in the way the numbering in Figure 9.3 records.

The designer stays in the loop. The bar below the capability ring, with arrows running both ways, marks the limit we place on the third step. Autonomy of design is not autonomy from oversight.

The system proposes a revision; a designer sets the bounds within which it may propose, authorizes the advance, and corrects the model where it is wrong. Explanation and audit belong among the capabilities rather than beside them for exactly this reason, since a designer can authorize only what the system can explain. The transition condition of Section 9.3.6 is an early form of such a mechanism, holding an advance against four criteria a designer states, and the designer's authority is something we mean to preserve through the escalation rather than surrender along the way.

From platform to program. Through a platform we make work cumulative rather than serial, and through the DSIM we offer three things by which the work compounds. There are the constructs, the tri-space characterization, the compiled aspiration, the cognitively grounded stakeholder agent, and the knowledge graph held as a formal state variable, each of which a designer instantiates again for a new domain rather than rebuilds. There is the evaluative apparatus, the eleven requirements of Chapter 2 and the Verification and Validation Square of Chapter 1, against which a later contribution is weighed on the same terms as an earlier one. And there are the open questions, which we state precisely enough for others to work on and bound against what researchers already achieve through established approaches.

A platform becomes a program when we pursue its questions together rather than apart, and in Figure 9.3 we show why these five belong together. The capabilities in the ring depend on one another. A designer cannot authorize a system that revises its structure without explaining the revision. One that revises its standard without anticipating the social response revises toward a target stakeholders will not accept. One that holds its context implicitly cannot expose a revision for a designer to correct. These are not five separate agendas that happen to share an origin. They are one problem decomposed: how to build an evolving system that redesigns itself and stays answerable while it does. That problem is the program, and the strategic engineering of Evolving Cyber-Physical-Social Systems is where we pursue it.

9.13 Perspectives on the Progression Revisited

In Chapter 1 we describe the progression of the dissertation along two axes, and we defer to this chapter the demonstration that the assembled framework advances along both at once. We return to them here. The two axes are not two descriptions of one progression. They are two independent progressions, and the DSIM is where they meet.

9.13.1 Capability Escalation

Along the first axis we escalate what a designer is able to do. We build from Observe through Structure, Adapt, Augment, Execute, Assess, and finally Synthesize, and we establish in Chapter 1 that this sequence is not an arrangement we choose but an order we take from the structure of the E-CPSS design problem.

We confirm the order through the assembled framework. We observe the state in Layer 1 before we structure decisions within it, because the decisions warranted depend on a characterization we do not yet hold until we produce it. We structure decisions in Layer 2 before we adapt across a transition, because we adapt by working on the decisions that connect one state to the next. We adapt before we augment, because augmentation extends a design process that must be structurally sound before there is anything worth extending. We augment before we execute, because a designer whose reach is extended still works at human speed, and where a system moves faster than human deliberation we must execute without waiting on it. We execute before we assess, because the social legitimacy of a strategy is a property of a strategy that exists, and there is nothing to assess before one is designed and made executable. And we synthesize last, because synthesis is the assembly of the six into a framework, which is the work of this chapter and which we can do only once we hold all six capabilities.

We map each capability to a layer, and we read the escalation as the vertical structure of the DSIM from the bottom upward, from observation at Layer 1 to assessment at Layer 5, with synthesis as the assembly we complete through the transition condition. Along the capability axis we answer the question of what a designer can do, and through the framework we escalate it from reading a state to authorizing an advance.

9.13.2 Knowledge Escalation

Along the second axis we escalate what a designer comes to know. We produce predictive knowledge, then structural, dynamic, generative, experiential, and simulated knowledge, and we hold each as an instance of K(si), tagged to the state at which we produce it and fed forward into the characterization of the next.

We confirm through the assembled framework that this axis is distinct from the first. Along the capability axis we address the action a designer takes at a state. Along the knowledge axis we

address the understanding a designer accumulates across states. A designer who advances along the capability axis alone would arrive at each new state able to act and knowing only what the current state presents. Along the knowledge axis we carry understanding forward, so that we draw the characterization of a later state on everything we produce at the earlier states. We establish in Section 9.3.5 that this forward feed, the connection of K(si) into I(si+1), is a property of the assembled framework rather than of any single method, and we read the knowledge axis as that property seen as a progression. By the sixth state a designer characterizes the system through predictive, structural, dynamic, generative, experiential, and simulated knowledge at once, none of which is available at the first.

Along the two axes we escalate different things. Along the capability axis we escalate the reach of action, layer by layer, within a state and across its transition. Along the knowledge axis we escalate the depth of understanding, type by type, as we carry the system from state to state. A framework we advanced along one axis alone would be incomplete in a way we name precisely along the other. A designer with the capabilities and without the accumulating knowledge would act at each state uninformed by the last. A designer with the knowledge and without the escalating capabilities would understand the system and be unable to act at the orders of complexity it presents. We advance the DSIM along both, and we synthesize them in the single framework, so that a designer both acts across the full arc of a state and its transition and carries forward the understanding we produce at each state. This is the convergence we defer from Chapter 1 to this chapter, and we show it in Figure 9.4.

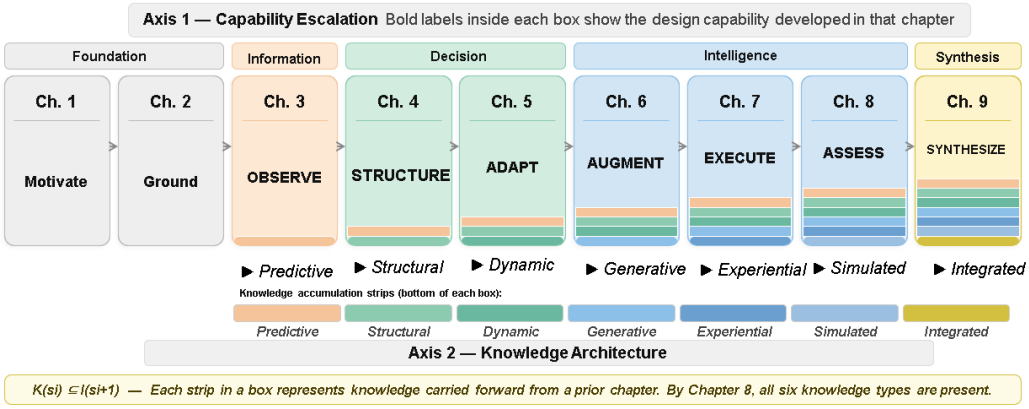


Figure 9.4: The Two Axes of Progression Converging on the Decision-State Information Model

9.14 I Statement

Throughout this dissertation I write in the first person plural. Here I write in the first person singular, because what follows is mine to claim and mine to be held to.

I organize this statement in three parts. In Section 9.14.1 I reflect on what I have learned and how I have grown. In Section 9.14.2 I assess myself against the competencies I set out to develop and describe where I intend to grow next. In Section 9.14.3 I set out what I intend to do with what I have learned.

I hold one idea across all three parts, and I state it at the outset. In Chapter 2 I argue that a designer who treats the cyber, the physical, or the social space of a system as an afterthought designs a system that falls short in that space. I hold the same idea about a career. I intend to contribute economically, intellectually, and socially, and I treat these as one commitment rather than three ambitions among which I will later choose. When someone creates real value through a method, I call that engineering. When others learn something durable from a venture and create new knowledge, I call that scholarship. And when the people I reach through a piece of work are glad (glad here could have certain perspectives) to receive it, I consider the work worth doing. I take the coevolution I argue for in the system and I apply it to myself.

9.14.1 Self-Reflection

9.14.1.1 Where I began

I came to the University of Oklahoma from Pune, India in August 2022 to pursue my doctorate with Professor Janet K. Allen and Professor Farrokh Mistree. I arrived with a bachelor's degree in Mechanical Engineering, and Industrial and Systems Engineering was entirely new to me. I had not formulated a decision support problem. I had not read about Herbert Simon. Above all, I had not encountered the idea that the way a designer arrives at a decision can itself be studied, formalized, and improved, quite separately from whatever design results from it. I had spent four years learning to design things. I had not considered that someone might study designing.

What I did bring with me was a strong pull toward systems and products. Through my undergraduate work I found that I cared less about any single component than about what happened

when I put components together, and I kept noticing that a machine I understood perfectly on paper behaved differently once it sat in a plant, surrounded by supply, schedule, cost, and the people operating it. I wanted to work on that gap. I followed that pull into Industrial and Systems Engineering, and I have not lost it since.

Between August 2022 and July 2026, I completed a Master's and Ph.D. in Industrial and Systems Engineering, and a graduate certificate in data science and analytics, in three years and eleven months. I record this because it says something about how I worked rather than because the timing matters in itself. I was eager to move, and I chose to keep moving. I formed positions while I was still building the vocabulary to express them, and I stayed willing to revise a position I had held for only a few months when I found a better one. I came to enjoy working that way, and I would choose it again.

9.14.1.2 The question I keep returning to

I have come to describe what I want from an engineered system in a single question, and I find it useful because I can hold it in mind while working on something complicated.

Can I give an engineered system a voice, so that it tells me what it knows, what it is aiming for, and where it is unsure? And can I answer back, correcting what it believes and adjusting what it aims for, so that the two of us hold a conversation rather than exchange inputs and outputs?

I followed this question toward adaptive systems, and then toward systems that redesign and correct themselves. Once I can converse with a system, I am most of the way to being able to improve it, because in the conversation I get both halves of what improvement requires: I can see what the system believes, and I can change what it is trying to do. Once I can do both, I can ask a harder question, which is whether the system can take over parts of that revision itself while I retain the authority to approve it. That progression, from a system I can interrogate, to a system that adapts, to a system that participates in its own redesign, is the arc I have been working along.

To hold that conversation, I need two things from the system. It must hold its knowledge in a form I can read, and it must hold its purpose in a form I can revise. Through Chapter 6 I make design knowledge into something a designer queries and interrogates. Through Chapter 7 I make the

standard for good enough into something I can trace back to the facts I set it from. Through Chapter 8 I model how the people on the receiving end of a decision respond, so that the conversation includes them as well. Reading the three together, I recognize that I have been building toward a system I can speak with, and I intend to keep building toward it through artificial intelligence and, in time, artificial general intelligence.

9.14.1.3 How I came to work on artificial intelligence

I did not arrive with an interest in artificial intelligence. I arrived with an interest in decisions, and artificial intelligence is where that interest led me.

In my first year I worked on how a designer decides under uncertainty, and I kept running into the same obstacle from different directions. Decision makers judge the quality of a decision partly by how accurately they understand the situation when they make it. An experienced designer carries that picture in their head: which supplier is unreliable this quarter, which regulation changed last month, which constraint the plant manager will quietly ignore, which stakeholder will object and why. In a formal decision model I capture almost none of it. So the designer gets a recommendation that is internally consistent and situationally naive, and either overrides it, wasting the model, or follows it, wasting the own judgment of the designer.

I wanted to close that gap, and to close it I have to find a way to put context inside the decision. That is what brought me to artificial intelligence. I did not go there because the field was prominent; I went because researchers in machine learning, knowledge representation, and reinforcement learning have built the tools for holding context and acting on it, and I needed those tools. In Chapter 3, I first structure the observation process so that designers interpret the knowledge I produce and use it in subsequent design decisions. In Chapter 6, I represent contextual knowledge in a form that designers interpret, evaluate, and correct. In Chapter 7 I compiled the standard the agent holds itself to from that context. I began each of these as a decision problem and finished with an artificial intelligence problem, because in each case I followed the context.

While I worked on this, the field itself moved a long way, and I found that genuinely exciting to watch. Over four years I watched a speculative question turn into an engineering question, and I watched capital and capability arrive at a scale I had not anticipated. In a 2025 industry survey of fifteen hundred business leaders across fourteen countries, all of them in large organizations

already exploring agentic AI, researchers project that organizations could generate as much as four hundred and fifty billion dollars in economic value through AI agents by 2028, through revenue growth and cost savings across the markets they surveyed [1]. In a 2026 industry survey of three thousand two hundred and thirty-five business and technology leaders across twenty-four countries and six industries, researchers find that close to three-quarters of companies plan to deploy agentic AI within two years [2]. From these I take it that the appetite question is settled, and that the gap now lies elsewhere.

In the same surveys I find where that work lies, and this is the part I find most useful. Thirty-four percent of companies report using artificial intelligence to deeply transform their business; three in ten are redesigning key processes around it; and thirty-seven percent report using it only at a surface level, with little or no change to the business processes underneath [2]. Among the companies planning to deploy agentic AI, twenty-one percent report having a mature model for agent governance [2]. And in the first survey, twenty-seven percent of organizations expressed trust in fully autonomous AI agents, down from forty-three percent twelve months earlier, with ethical concerns, lack of transparency, and limited understanding of what these agents can do identified as the key barriers [1].

I find that last pairing the most instructive result available to me. Over roughly the same period, researchers record organizations broadening workforce access to artificial intelligence from under forty percent to around sixty percent of workers, and twice as many leaders as a year before reporting a transformative effect on their company [2]; and they record only twenty-seven percent of organizations expressing trust in fully autonomous agents, down from forty-three percent [1]. I draw these from two separate surveys with different respondents, so I hold the comparison loosely. Even so, I conclude from it that the constraint is no longer how much a system can do. The constraint is whether an organization can say what its system is aiming for, show why it decided what it decided, and be reasonably confident that the people affected will accept the result. Purpose, transparency, and legitimacy are the three things in short supply, and they are the three things I spent Chapters 6, 7, and 8 building mechanisms for.

This is what I mean when I say I want to work on artificial intelligence with purpose. I use the phrase as a specification rather than a sentiment. I consider a system to be pursuing a purpose when I can name the standard by which it judges its own performance, show where that standard came

from, revise it as the world changes, and defend it to the people it affects. I have built machinery for each of those four, and I want to spend the coming years building better machinery for all four.

9.14.1.4 Four ways my thinking developed

I can name four ideas I now hold that I did not hold when I began.

I now treat the social space as a design space rather than an acceptance problem. Consider a consumer goods company preparing to release a redesigned product. Engineering, sourcing, and pricing are settled months in advance, and the question of how customers will actually receive the product gets answered after launch, when changing course is expensive and reputations are already committed. I want a designer to be able to ask that question while the design is still movable: to run the release in simulation, watch a population of customers with different priorities and different tolerances respond to it, see which of them turns away and over what, and then revise the product rather than the messaging. I began by treating stakeholder response the way engineers commonly treat it, as something to manage once the technical design is settled. I now hold that a technically excellent intervention succeeds only when the people it reaches are willing to work with it, and that I therefore owe the social space the same modeling care I owe the physical one.

I now treat satisficing as a theory of action rather than as a compromise. Consider a plant manager scheduling next week's production against a demand forecast. By the time she gets the best schedule back from an optimizer, a supplier has slipped by two days, a customer has amended an order, and a technician has taken a machine down for unplanned maintenance. That optimum was real when someone computed it and is no longer real when she has to execute it, and she is left holding a precise answer to a question that has since changed. I inherited satisficing and bounded rationality principles from the tradition of our laboratory, the Systems Realization Laboratory at the University of Oklahoma, and at first, I understood it the way most people first do, as accepting less. I now hold that when the conditions of a problem move, the optimum is not a target I am failing to reach so much as a target that will not stay still long enough to be worth chasing, and that a solution robust across the range the conditions will actually take is the better object to pursue.

I now treat context as a structure rather than as an input. Consider a demand model recommending reorder quantities inside a procurement system, and a sourcing rule that changed

two months ago. Someone fitted the model well, its recommendation is now wrong, and nobody can see why, because nowhere in that model did anyone write down what it assumes about sourcing. The assumption is real, it is load-bearing, and it is invisible. If instead a designer holds those assumptions as statements she can read, question, and edit, she can make the correction in an afternoon rather than a quarter. I began by treating context as a conditioning vector supplied to a model. I now hold that for an evolving system, context is most useful when I keep it as an inspectable, editable structure, and that what it should set is the standard for good enough rather than the reward or the utility scale.

I now treat functionality as the test rather than novelty. Consider two proposals addressing the same problem. In the first, I would apply a technique nobody has applied in this setting before. In the second, I would give the designer something they currently cannot do at all. Early on I would have found the first more attractive, because I judged my own ideas by asking whether anyone had done them before. I have learned to prefer the second, and to be a little suspicious of my own enthusiasm for the first. When I identify a functional gap correctly, I find that I get novelty along with it. I would call this a small adjustment, and yet after making it I read papers differently, chose problems differently, and recognized differently when an idea was ready.

9.14.1.5 The question I carry forward

I have kept my attention on one question throughout this work, and I expect it to hold my attention for a long time yet: **how do I represent context structurally, and how do I choose the right context at a given state and under a given set of priorities?**

I have answered parts of it here. In Chapter 3 I structure observation across three spaces before I train anything, so that what I recover afterward is interpretable for design rather than only accurate. In Chapter 6 I validate each candidate knowledge statement quantitatively before I admit it to the Knowledge Graph, so that the context I carry forward is context I have earned. In Chapter 7 I treat the Knowledge Graph as a formal state variable and compile the agent's aspiration from it under stated axioms, so that context sets the standard rather than the reward. In Chapter 5 I answer a version of the priority question across time, by holding transitions to conditions at every state rather than at the endpoint alone.

I reached these answers by trying several approaches and taking a specific lesson from each. When I represented context as additional features appended to a model, I learned that a larger input does not by itself produce more understanding, and that I have to impose structure before training rather than recover it afterward. When I represented context as a quantity to be inferred, I learned how much I valued being able to inspect and correct what the system believed, and that lesson is why I now keep context in an explicit and editable form. And in working through how different stakeholders bring different priorities to the same context, I learned that selection and prioritization are one problem rather than two, which is why I give the weight scenarios in Chapters 4 and 5 as much work to do as I do.

I carried each of these lessons directly into the design of the next method. Taken together, I read them as marking where the remaining answer lies, and I set out the specific research directions in Section 9.12.

9.14.1.6 Deciding with incomplete information

If I name the capability I most clearly developed here, it is the ability to reach a decision inside a new method when the information available to me is incomplete, and to be clear afterward about how I reached it.

In every method I combine two things. I draw on what I take from the literature, which I read carefully and which constrains a great deal, and I rely on my own judgment past the point where I can take an answer from the literature. I chose which spaces to structure observation into. I chose the form of the coupling in the decision model. I chose the axioms an admissible compiler should satisfy. I chose to make arousal the quantity that scales how strongly emotional state biases retrieval. In each case I found that the literature narrowed the space of reasonable options without selecting among them, and the question in front of me was simply: how do I proceed?

The way I proceed is to ask what the fundamental issue is and where the fundamental gap lies, and then to ask whether I can turn that gap into a research question worth answering. I get most of my traction from that reframing. I can often make progress on a question I could not answer directly, once I stop asking which option is correct and start asking what is actually missing and for whom. Then I make the judgment, I state clearly where I made it, and I build on it so that whoever comes next can examine it and improve on it. I used to look for a derivation that would settle these

questions for me. I now regard the willingness to decide, together with the discipline of being transparent about the decision, as one of the more valuable things I take from this work.

9.14.1.7 The idea I expect to carry furthest

Through each of the six methods I answer the challenge for which I developed it, and I need all six together to make the argument of this chapter.

If I name the idea I expect to still be working on in twenty years, it is this: **take what a designer would ordinarily leave inside a model and put it outside, where someone can read it, edit it, and argue with it.** I applied that idea twice, to unrelated problems, before I noticed I was doing it, and noticing it changed how I think about what I am building.

Stated as a capability, through this idea I get a system whose beliefs and whose standards I can both correct. I can ask it what it thinks is true and change my mind or its mind. I can ask it what it is aiming for and adjust the aim. I can do neither of these when both sit inside weights.

Stated as an opportunity, this is where I see the largest gap between what operating organizations have and what they need. Today, the person accountable for a system's decisions cannot reach most of what that system holds. That person cannot inspect it, cannot correct it, and therefore cannot in good conscience authorize it to act. I read the decline in organizational trust I describe in Section 9.14.1.3 as this gap showing up in a survey. I see closing it as a technical problem with a clear commercial consequence, and I find that an unusual and attractive combination.

Stated as a path forward, I take from this idea a program rather than a project. When I externalize what a system believes, I arrive at world models an operator can govern. When I externalize the standard it holds itself to, I arrive at a theory of how context sets what counts as good enough. When I externalize the mechanisms of judgment, I arrive at cognitive architectures whose behavior I can examine rather than infer. I take up all three in Section 9.14.3.

9.14.1.8 How I have grown

I came into this program expecting to leave it an expert. I am leaving with something I have come to value more.

I have learned how to approach a research gap in an area that is new to me. This is the change I notice most often in practice. When I meet an unfamiliar field now, I know how to find its boundary, how to read enough of it to state what its researchers already achieve, and how to locate the place where something useful is missing. I did this repeatedly across this dissertation, in reinforcement learning, in cognitive science, and in knowledge representation, and in each case I began outside the field. Knowing that I can do this again matters considerably.

I have learned how to think toward a goal when there is no established right answer. This took me the longest to accept. Early on I looked for the correct way to proceed, as though one existed and I had simply not found it yet. What I have learned is that in most research worth doing there is no single right path, and that the useful skill is to hold a clear picture of where I am trying to arrive and then reason toward it, choosing among reasonable options, checking whether each step brings me closer, and revising when it does not. Thinking with a destination in mind, rather than waiting for the right method to present itself, is how I now work.

I have learned what I do not know, and how to find a path toward knowing it. I now hold that a doctorate is less about accumulating expertise than about developing an accurate picture of one's own ignorance. But the picture alone would not be worth much. What makes it a competency is the second half: knowing how to find a route into unfamiliar ground, whom to read, whom to ask, what to build first, and how to tell whether I am making progress. I regard this as the single most important thing I take from these four years, and as the competency I expect to rely on for the rest of my working life, because at some point in every problem I care about, I will have to learn something I do not currently know. I expect whatever expertise I hold to age. I do not expect my ability to acquire more of it to age with it.

I have developed a clearer sense of myself. I came in still forming a sense of who I was as a professional, and I am leaving with a clearer one. I have a better sense of which problems interest me and which I would decline. I find it easier to hold a position when it is questioned, and easier to let one go when the questioning is right, and I have come to think those are the same skill rather than opposite ones. More than that, I have found that in shaping the work I was also shaping myself. In choosing which questions to pursue, whose criticism to take seriously, and what I would not compromise on, I was also deciding what I value and who I intend to be. I did not expect a

doctorate to be an exercise in self-definition, and I now think that is among the truest things about it. I still have a good deal to learn in all of this, and I am comfortable saying so.

9.14.1.9 Janet, Farrokh, and the Systems Realization Laboratory

I owe most of that growth to Professor Janet K. Allen and Professor Farrokh Mistree.

What they gave me, consistently and generously, was the encouragement to become an independent researcher. They could have handed me a topic and a method. Instead they gave me the room to find my own, stayed close and engaged while I found it, and let me work through my own reasoning. I understand now what that cost them in patience, and I am grateful for it.

Much of what I take from them is not just technical. They showed me that a mentor can be warm and available to a student while holding the relationship to a high professional standard. They showed me how to empower someone rather than direct them. They showed me how to build a person's confidence through steady individual attention, which I received from both of them and intend to offer others. They took care that each person around them developed an identity of their own rather than a version of theirs, and I now see how difficult and how rare that is. And they asked, in every discussion, what the work was finally for, and would not let technical excellence stand as an answer by itself. That habit is the one I most want to carry with me.

9.14.2 Self-Assessment Against Research Competencies

I set out to develop a set of career-sustaining competencies: to keep learning through reflection on doing, to identify gaps that lead to new knowledge, to ask questions and listen carefully, to decide with incomplete information, and to think critically and speculate a way forward.

I assess myself against some selected competences in Table 9.9. In the third column I describe how I intend to build on each.

Table 9.9: Self-Assessment Against the Competencies I Developed

Competency	How I developed it in this dissertation	Looking forward

Identifying a research problem by defining a boundary	I define the E-CPSS as the design object, identify information asymmetry and intelligence asymmetry as the sources of difficulty, and derive six escalating design challenges from them.	I have learned to set a boundary from the structure of a field. I will extend the same skill to setting a boundary from what an organization needs enough to invest in.
Posing questions worthy of investigation	I map seven research gaps to a principal research question and four secondary questions, and in Section 9.12 I bound five further avenues against what researchers in established fields already achieve.	I bound questions well against literatures. I will learn to bound them equally well against markets, so that the questions I pursue next are useful in both senses.
Deciding with incomplete information	In every method I combine what I take from the literature with judgments I make where the literature runs out, and I state each judgment where I make it.	I make these decisions with time to reflect. I intend to develop the same clarity at operational speed, where a decision affects a team and a customer.
Proposing and executing a plan	I developed six methods over four years, answered a stated question with each on a real test problem, and assembled them into one framework here.	I have executed a plan of my own design. I will build on that by planning work that others execute alongside me, which is what leading a group requires.
Establishing validity and articulating new knowledge	I organize the dissertation around the Verification and Validation Square, close all four quadrants in Sections 9.7 and 9.8, and state each contribution as a new knowledge statement.	I have established validity in the four quadrants a dissertation reaches. I intend to extend the same rigor to Establishing that someone will invest in what I build.

Building a working system	I implemented the generative framework of Chapter 6, the learning agent of Chapter 7, and the twenty-agent cognitive simulation of Chapter 8.	Through these systems I demonstrate each idea convincingly. I will build on that toward systems engineered for reliability, cost, and continuous operation.
Carrying a construct across domains	I demonstrate six methods across six unrelated domains and, in Section 9.8, separate the domain-specific inputs of each method from its domain-independent structure.	I make the invariance argument structurally, and I look forward to strengthening it empirically by carrying a construct into a domain a collaborator chooses rather than one I choose.
Communicating scholarship	I have presented this work at IDETC, IISE, and INFORMS, and prepared the manuscripts listed in Section 9.14.5.	I communicate comfortably with technical audiences. I will develop the same fluency with the operations, policy, and business audiences who decide whether work like this gets used.
Teaching and mentoring	I co-orchestrated Design for Open Innovation and Systems Engineering, and mentored approximately eight undergraduate and master's students.	I found this work more rewarding than I expected. I intend to keep teaching and mentoring in whatever setting I work in, and to take on more responsibility for it over time.

9.14.2.1 Where I intend to grow next

Reading the third column of Table 9.9 as a whole, I see that I have written every entry in the same direction, and I want to name that direction plainly, because I have shaped my plans for what comes next around it.

Establishing that the work creates value someone will invest in. I organize this dissertation around the Verification and Validation Square and close all four of its quadrants. I have come to see a further form of validity lying outside what a doctoral dissertation is designed to reach, and I think of it as economic validity: whether an organization will adopt a method, invest in it, and continue using it once it is no longer new. This is not a shortcoming of the work; it sits beyond the scope of the instrument, as it does for doctoral research generally. I find it a compelling question, and I am drawn to it because through the training I received here I am equipped to ask it rigorously. Learning how to establish this form of validity is the single thing I most want to do next.

Building and leading a group. Through mentoring students inside a laboratory where I already had the structure, the funding, and the culture around me, I have learned a great deal about how people learn and how much steady encouragement matters. Creating those conditions myself is the next thing, and I look forward to it: bringing people together, taking responsibility for their work and their growth, and holding a group steady through the periods when progress is slow. I expect to learn this mainly by doing it.

Deploying in the field. I draw my test problems from real design situations, and I examine each retrospectively or in simulation. What I look forward to is handing a method to a practitioner and watching what happens when the outcome matters to her, because I expect to learn things about my own methods that I could not learn through any amount of further analysis. I regard this as one of the most informative experiences available to me.

Engineering for continuous operation. Through my systems I demonstrate each idea convincingly at the scale I built them for. Carrying them to production standard, where I have to account for reliability, latency, cost per decision, and graceful degradation, and where I am not there to watch, calls for a distinct engineering discipline, and it is the one I most want to acquire next. Together with field deployment, this is what I mean by translational capability, and I take it up directly in Section 9.14.3.8.

Speaking to the people who decide. I can defend every construct here to a technical audience. The people who decide whether a method is adopted are frequently operations directors, policy officials, and business owners, and they will judge the work by whether I can explain what it does for them in language that costs them nothing to follow. I regard this as a skill to be practiced deliberately, and I intend to practice it.

I record these five as an inference I draw from my own record rather than as a list of shortfalls. In each I name a foundation I laid here and a natural next step from it, and from them taken together I can explain why I have chosen the path I describe in the section that follows.

9.14.3 My Vision: Engineering Intelligence

9.14.3.1 The question I intend to work on

How do I build engineered systems that hold an explicit, inspectable model of their own world and obligations, that recognize when that model has drifted from reality, and that revise their own structure, goals, and standards in ways that remain accountable to the people they affect?

This is the formal version of the question I pose in Section 9.14.1.2. I laid its foundations here without knowing at the outset that this was the question, and I intend to work on it over the coming years.

9.14.3.2 Three positions I hold

I state three positions. Each is open to challenge, and I hold each one because of what I found while building.

I hold that the binding constraint on artificial intelligence in the economy is purpose, transparency, and legitimacy rather than capability. I take this from the evidence I present in Section 9.14.1.3, where I read growing capability alongside declining trust in autonomous systems. I do not expect a larger model on its own to close the gap between the capital deployed and the transformation realized. expect organizations to close that gap when their systems can state what they are aiming for, show how they decided, and win the acceptance of the people they touch.

I hold that context is most useful when it sets the standard rather than the reward. In nearly all the work I know that conditions behavior on context, researchers condition the utility function, the reward, or the transition dynamics. I hold that for a system operating in a world that moves, the object worth conditioning is the aspiration, the threshold for what counts as good enough. I

formalize one instance in Chapter 7, and I regard the general claim as the most transferable idea I offer in this dissertation.

I hold that I do better to place the mechanisms through which I make a system trustworthy outside the model rather than inside it. A designer who notices that a world model held only in weights is wrong cannot correct it. An observer cannot readily distinguish an emotional response produced by prompting from one that is performed. I hold that inspectability is a structural commitment I make early, and that I decide from this commitment what I can trust the system to do.

I pursue three thrusts from these positions.

9.14.3.3 Thrust 1: Context as a First-Class Object in Reinforcement Learning and Decision Theory

The questions I ask. I approach this as a progression of three.

First, how do I decide which part of the world a system needs to attend to at a given moment? A system knows more than I need for the decision in front of it, and by selecting well I get context that informs the decision rather than context that dilutes it.

Second, how do I make that selection scale? In a large system I face a large description of the world, I have to decide quickly, and I have to make the selection over and over. A selection rule that is principled and expensive is of limited use in operation, so I treat scalability as part of the problem rather than as an implementation detail to handle afterward.

Third, what does a normative theory of choice look like when the context governing a decision is a structured artifact the decision-maker revises, and when what that context sets is the aspiration rather than the utility scale? I arrive at this theoretical question from the first two, and by answering it I would turn a working construct into a general account.

Why I take this up now. Researchers in contextual reinforcement learning represent context as a latent vector inferred from history, and devote much of their effort to the inference. Researchers in decision theory represent it as a state index over utilities or as a reference point formed from expectations. In both, they condition the scale on which outcomes are valued. Organizations are

now deploying agentic systems into settings where the rules change during operation, and I want to know how well I can serve that setting with either representation.

The new knowledge I intend to establish. A theory of decision under evolving context in which the aspiration is the conditioned object, together with scalable methods for selecting the context that sets it, and guarantees that hold when the underlying structure is imperfect.

What I build on. The formalism I develop in Chapter 7, the satisficing commitment I make in Chapters 4 and 5, and the open questions I state as Avenues 1, 2, and 3 in Section 9.12.2.

Where I begin. With the context-selection problem at scale, because I see it as the immediate obstacle to using the construct anywhere the description of the world is large.

9.14.3.4 Thrust 2: Governable World Models

The questions I ask. Can I build a single inspectable and editable structure that I can use at once as the world model an agent acts against, as the artifact a designer audits and corrects, and as the shared grounding for a population of interacting agents? How much of an agent's behavior can I safely rest on such a structure when parts of it are wrong? And how do I get a system to explain not only a single decision but a proposed change to its own design?

Why I take this up now. Researchers hold established world models either implicitly, in weights no designer reads, or explicitly, as a structure built for the agent's own planning rather than for anyone's inspection. In established methods for governing an agent, researchers hold it to a symbolic specification that is fixed rather than editable. An operator asks a different question: what does this system believe, and what happens when I correct it? From the decline in trust I report in Section 9.14.1.3, I take it that many people are asking that question and few are receiving an answer.

The new knowledge I intend to establish. A construct and accompanying theory for structured knowledge I can use simultaneously as the agent's world, the designer's audit surface, and the shared social grounding, with stated bounds on the cost of error.

What I build on. The Knowledge Graph as I use it in Chapters 6, 7, and 8, the graph-perturbation robustness result I prove in Chapter 7, and Avenue 5 of Section 9.12.2.

Where I begin. With a single working governable world model in an operational setting, and a documented instance in which an operator corrects the model and I measure the effect downstream.

9.14.3.5 Thrust 3: Cognitive Architectures for Machine Judgment

The questions I ask. Do I do better to treat emotion, memory, and arousal-as-epistemic-uncertainty as a deterministic substrate beneath an action-selecting model than to elicit them from within one? What does bounded rationality look like when I implement it as a mechanism rather than assume it? And how closely can I get a cognitively grounded simulation to anticipate what people actually do?

Why I take this up now. Researchers are building multi-agent simulations populated by language models at a rapid pace, and I want to be able to trust them as evidence. That requires me to hold emotional behavior as a mechanism I can inspect rather than a label the model was asked to produce, and to measure departures from rational self-interest rather than assert them. As organizations begin to inform real decisions about policy, markets, and operations with such simulations, I expect that distinction to matter a great deal.

The new knowledge I intend to establish. A calibrated cognitive substrate for machine judgment, with measured departures from rational-actor prediction, and an account of the mechanism linking arousal to both inferential precision and memory.

What I build on. The architecture I develop in Chapter 8, the capability-ladder methodology I introduce there, and Avenue 4 of Section 9.12.2.

Where I begin. With calibration against human behavioral data, because every stronger claim I want to make depends on it.

9.14.3.6 Why I treat these as one problem

I could pursue any one of these three alone and produce respectable work. I intend to pursue them together (and maybe there are more), because I have come to see them as one problem separated into parts.

Were I to let a system revise its own standards without giving it an inspectable world model, I would have built something no one can examine. Were I to give it that world model but no account

of how the people it affects will respond, I would have aimed it at a target those people will not accept. And were I to model social response without a principled theory of how context sets the standard, I would have left the system nothing to act on once it learns something. I therefore treat the three as mutually supporting, and I show the shape of that dependence in Section 9.12.3.

9.14.3.7 On general intelligence for engineered systems

I now hold a considered view on artificial general intelligence, and I hold it with interest rather than certainty.

I believe general intelligence in engineered systems is achievable, and I believe the route to it runs through understand more about the directions I speak about. The generality I care about is a system's capacity to enter a situation it was not designed for, form a model of that situation it can expose, derive from that model a standard for acceptable performance, act against that standard, notice when the model has drifted, and propose a revision it can explain to someone with the authority to approve it. A system that performs many tasks well and can do none of that is broad rather than general, and I read the gap between investment and realized transformation I describe in Section 9.14.1.3 as evidence that breadth alone does not convert into value.

I place my own contribution honestly within that picture. In Section 9.12.3 I show that through this framework I revise the goals and the standard while the structure stays fixed, and I therefore place this dissertation between a self-adapting system and a self-designing one. I offer a foundation rather than an arrival, and I find the distance between them exciting rather than discouraging.

I hold this view as a working hypothesis and I intend to keep testing it. I will take it as confirmation that I am on a useful route when I find that systems I build with explicit world models, compiled standards, and grounded social anticipation prove more reliable in operation and more readily trusted than systems of comparable capability built without them. That is a comparison I can construct, and I intend to.

I want to be clear about one commitment, because it matters to me more than the position itself. Autonomy of design is not autonomy from oversight. In everything I intend to build, I keep a person in the loop with the authority to set the bounds, approve the advance, and correct the model. I let a system propose, and I reserve the deciding for a person. I hold this as a design requirement rather than a concession, and I intend to keep it as capability grows.

9.14.3.8 How I intend to take this to the world

In the five directions I name in Section 9.14.2.1, I point at one thing. In them I name the competencies I would need to carry a fundamental result across the distance separating it from something a practitioner uses and pays for, and I have a name for that distance: translational research. I have chosen my next years around learning to cross it.

Through this dissertation I did fundamental research. I developed methods, proved results, and established validity in four quadrants. In translational work I would take such results deliberately toward use, and I now regard that as demanding a discipline of its own. I would call a method applicable only once someone has engineered it for continuous operation, deployed it where the outcome mattered, learned how it behaves with real users, explained it to the people who authorize its adoption, and then rebuilt it. Each step in that sequence is one of the five directions I named, and I did not arrive at that list by accident.

What I intend to do. I intend to become an entrepreneur and, over time, an industrialist to create economic, intellectual, and social value. I want to build a company working at the frontier of engineering intelligence: an organization that does research, advises on hard problems, and turns what it learns into products and platforms others build on. I keep the specific idea open. I will consider doing what I intend to by becoming a professor, owing to my interests in researching and mentoring.

What I do hold is a clear view of where the value sits and can be further developed. Through this dissertation I develop the components of a platform for designing adaptive, self-correcting, and eventually self-designing systems: a way to structure observation, a way to formulate coupled decisions robustly, a way to work backward from a target state, a validated knowledge structure a designer queries, an agent whose standard I compile from that structure with guarantees, and a means of anticipating social response before deployment. Assembled and engineered properly, I believe I can turn those into infrastructure organizations build on rather than a study they cite. I wish to develop research, consulting, products, and platform alongside one another, and I expect what I learn from the organizations I work with to determine the sequence.

Where I will look. I am drawn to entrepreneurship: to agriculture, to manufacturing, to supply chains, and to finance, where people grow, make, move, and fund things, where conditions shift

faster than anyone can answer them through a design cycle, and where I can measure an improvement in the world rather than only on a benchmark. Beyond those, I am drawn wherever the frontier problem is the one I describe in Section 9.14.3.2, which is an organization with capability but without purpose, transparency, or legitimacy. I intend to build in industries rather than only for them.

What I will continue. I look forward to continuing to contribute to knowledge alongside all of this. I state the three thrusts above precisely because I want others to work on them. If I build well, I expect to run into questions in the building that I could not have posed from inside a laboratory, and I intend to publish them. I want a career in which I move in both directions between the fundamental and the applied, and I expect to do better in each direction because of the other.

9.14.4 Closing

I return to where I began.

In August 2022, a few weeks into a program in a field that was new to me, I went to Professor Janet K. Allen and Professor Farrokh Mistree with the only question I had, which was: what should I do?

They gave me something better than an answer. They told me that they could tell me what to do now, but that in the future I would be the one who has to do independent research, take responsibility for my life, and lead it in my own way.

I have grown into a fuller appreciation of that moment with every year since. At the point where I most wanted direction, they chose to give me the capacity to find my own. From that day, that question never troubled me again.

9.14.5 Publications Based on This Dissertation

Monograph

Bhalerao, M.J, Allen, J.K. and Mistree, F., “Architecting Adaptive Decision Intelligence for Evolving Cyber-Physical-Social Systems: A Strategic Engineering Approach to Knowledge-Driven Adaptive Design,” Proposal accepted on July 30, 2026.

Journal Papers

1. Bhalerao, M.J, Tillari, D.K., Milisavljevic-Syed, J., Allen, J.K. and Mistree, F., "Designing Adaptive Requirements for E-CPSS through LLM Based Reasoning for Change Propagation Analysis" [Manuscript in preparation; Target Date: August 15, 2026].

2. Bhalerao, M.J., Mandegari, S., Allen, J. K., and Mistree, F., "Compiled-Aspiration Risk-Sensitive Q-Learning: Sourcing the Satisficing Threshold from a Structured Knowledge Graph," [Manuscript in preparation; Target Date – July 30, 2026].

3. Bhalerao, M.J., Kumar, K., Allen, J. K., and Mistree, F., "A Cognitive Architecture for Stakeholder Deliberation: Affect, Memory, and Neuro-Symbolic Reasoning," [Manuscript in preparation; Target Date – July 30, 2026].

4. Mandegari, S, Bhalerao, M.J, Allen, J.K. and Mistree, F., 2026, "Evolving Temporal Knowledge Graphs Through Simulation-Informed LLM Reasoning: A Closed-Loop Framework for Complex System Understanding," [Manuscript in preparation; Target Date: August 15, 2026].

5. Mandegari, S., Bhalerao, M. J., Allen, J. K., & Mistree, F., 2026, "Generative AI for Interpretable Satisficing Solution Design in Manufacturing," ASME Journal of Computing and Information Science in Engineering, 1-25. https://doi.org/10.1115/1.4071473

6. Gupta, H., Bhalerao, M. J., Honeycutt, W. T., Allen, J. K., and Mistree, F., 2025, "On Designing Evolving Cyber–Physical–Social Systems: A Decision-Based Design Perspective," ASME Journal of Computing and Information Science in Engineering; 25(12): 120803. https://doi.org/10.1115/1.4070094

7. Bhalerao, M.J, Mandegari, S, Allen, J.K. and Mistree, F., 2025, "Designing Robust Policies for Evolving Cyber-Physical-Social Systems in Uncertain Environments," ASME Open Journal of Engineering [Under Review].

8. Mandegari, S., Bhalerao, M. J., Allen, J. K., & Mistree, F., 2026, "Goal-Oriented Inverse Design for Robust and Adaptive Policies in Evolving Cyber–Physical–Social Systems," ASME Open Journal of Engineering, 5. https://doi.org/10.1115/1.4071373

9. Haynes, J. P. E., Bhalerao, M. J., Honeycutt, W. T., Allen, J. K., & Mistree, F., 2025, "Engineering Informatics for Aviation Safety: Machine Learning-Based Prediction of Bird Strikes Using a Model-Based CPSS Design," *ASME Open Journal of Engineering*, *5*. https://doi.org/10.1115/1.4070542

Book Chapters

1. Bhalerao, M.J., Honeycutt, W.T., Allen, J.K. and Mistree, F., 2025, "Utility of Function-Behavior-Structure Construct in Designing Public Policy: A Conceptual Exposition", In Responsible and Resilient Design for Society, Springer Nature.

2. Mandegari, S., Bhalerao, M. J., Allen, J. K., and Mistree, F. (2025). Designing a Business Plan for Sustainable Resilience: A Conceptual Exposition. In Responsible and Resilient Design for Society: Vol. 12 (pp. 125–138). Springer Nature. 10.1007/978-981-96-5511-2-10

Conference Papers

1. Bhalerao, M. J., Tillari, D. K., Milisavljevic-Syed, J., Allen, J. K., and Mistree, F. (2026). *LLM-based requirements evolution for evolving cyber-physical-social systems: A change propagation framework* [Paper presentation]. ASME 2026 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC-CIE 2026), United States. Paper No. DETC2026-194374.

2. Mandegari, S., Bhalerao, M. J., Mistree, F., Naidu, J. P., and Kataria, D. (2026). *Digital twin for crop health monitoring using quantum image representation* [Paper presentation]. Fourth International Conference on Industry 4.0 and Advanced Manufacturing (I-4AM 2026), Indian Institute of Science, Bengaluru, India.

3. Bhalerao, M. J., Mandegari, S., Allen, J. K., & Mistree, F. (2025). Designing Robust Policies for Evolving Cyber-Physical-Social Systems in Uncertain Environments. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 89213, p. V02BT02A059). American Society of Mechanical Engineers.

4. Mandegari, S., Bhalerao, M. J., Allen, J. K., and Mistree, F. (2025). From goals to actions: Designing policy as an evolving cyber-physical-social system. In *Proceedings of the ASME 2025 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Article V02AT02A034). American Society of Mechanical Engineers. https://doi.org/10.1115/DETC2025-167668

5. Sajid, Z. S., Bhalerao, M. J., Allen, J. K., and Mistree, F. (2025). Towards engineering solutions to light pollution and avian mortality: A system dynamics approach for sustainable urban development. In *Proceedings of the ASME 2025 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Vol. 3B, Article V03BT03A012). American Society of Mechanical Engineers. https://doi.org/10.1115/DETC2025-169594

6. Bhalerao, M. J., Honeycutt, W. T., Allen, J. K., and Mistree, F. (2025). Utility of the function–behavior–structure construct in designing public policy: A conceptual exposition. In *Research into design* (ICoRD proceedings). Springer. https://doi.org/10.1007/978-981-96-7308-7_10

7. Mandegari, S., Bhalerao, M. J., Allen, J. K., and Mistree, F. (2024). Predicting future states of evolving systems. In *Proceedings of the ASME 2024 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Vol. 3B, Article V03BT03A007). American Society of Mechanical Engineers. https://doi.org/10.1115/DETC2024-146340

8. Haynes, J. P., Bhalerao, M. J., Honeycutt, W. T., Allen, J. K., and Mistree, F. (2024). The impact of artificial lights at night (ALAN) on bird strikes: Predictive modeling for public policy design. In *Proceedings of the ASME 2024 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers. https://doi.org/10.1115/DETC2024-143780

9. Bhalerao, M. J., Honeycutt, W. T., Das, A. K., Allen, J. K., and Mistree, F. (2023). Framing wicked problems through evidentiary and interpretative analysis. In *Proceedings of the ASME 2023 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference: Vol. 3B. 49th Design Automation Conference*

(DAC) (Article V03BT03A005). American Society of Mechanical Engineers. https://doi.org/10.1115/DETC2023-117285

References for Section 9.14.1.3.

[1] Capgemini Research Institute, 2025, "Rise of Agentic AI: How Trust Is the Key to Human-AI Collaboration," Capgemini, Paris, France, from https://www.capgemini.com/insights/research-library/ai-agents/

[2] Deloitte AI Institute, 2026, "State of AI in the Enterprise: The Untapped Edge," Deloitte Development LLC, New York, from https://www.deloitte.com/us/en/about/press-room/state-of-ai-report-2026.html

Disclosure: AI tools were used to improve the readability of the dissertation.