

UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

**PROBLEM FRAMING BEFORE SATISFICING DECISION MAKING IN
RURAL CYBER-PHYSICAL-SOCIAL SYSTEMS**

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By SURAMYAA SARKAR

Norman, Oklahoma

2026

**PROBLEM FRAMING BEFORE SATISFICING DECISION MAKING IN
RURAL CYBER-PHYSICAL-SOCIAL SYSTEMS**

A THESIS APPROVED FOR THE
SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Janet K. Allen, Chair

Dr. Farrokh Mistree

Dr. Sesh Commuri

Dr. Doyle Dodd

©Copyright by SURAMYAA SARKAR 2026
All Rights Reserved.

ACKNOWLEDGMENTS

I am deeply grateful to my advisors, Professor Janet K. Allen and Professor Farrokh Mistree, for their guidance, patience, and rigorous feedback throughout this research. Their insistence on structural clarity and precise reasoning shaped not only this thesis but how I think about engineering design as a discipline. I could not have asked for better mentors.

I am grateful to the graduate faculty and my fellow students in the School of Industrial and Systems Engineering at the University of Oklahoma for their support, discussions, and camaraderie throughout my program.

Finally, I thank my family and friends for their patience, encouragement, and support throughout this journey.

TABLE OF CONTENTS

Level	Heading	Page
Chapter	CHAPTER 1: FRAME OF REFERENCE: PROBLEM FRAMING BEFORE SATISFICING DECISION MAKING	1
Section	1.1 Problem Context	3
Section	1.2 Problem Statement	5
Section	1.3 Research Gap	6
Section	1.4 Research Objective	7
Section	1.5 Rationale for Satisficing	8
Section	1.6 DTM-Based Problem Space Formulation	10
Section	1.7 Thesis Contributions	11
Section	1.8 Design Functionalities	12
Section	1.9 Design Functionalities to Development Outcomes	13
Section	1.10 Research Gaps	17
Section	1.11 Design Functionalities, Method Elements, and Development Outcomes	19
Chapter	CHAPTER 2: LITERATURE REVIEW: RURAL CYBER-PHYSICAL-SOCIAL SYSTEMS, SYSTEMS ARCHETYPES, AND SATISFICING DECISION FORMULATIONS	25
Section	2.1 Scope of the Review	25
Section	2.2 Rural Cyber-Physical-Social Systems	26
Subsection	2.2.1 The CPSS Paradigm and Its Rural Extension	26
Subsection	2.2.2 Resource Dynamics and Feedback in Rural Service Systems	27
Section	2.3 Systems Archetypes and Structural Feedback Interpretation	28
Subsection	2.3.1 Origins and Qualitative Formulation	29
Subsection	2.3.2 System Dynamics and Quantitative Simulation	29
Subsection	2.3.3 The Gap Between Archetype Identification and Decision Formulation	30
Section	2.4 Satisficing Decision Formulations and the Compromise Decision Support Problem	30
Subsection	2.4.1 The Satisficing Philosophy	31

Level	Heading	Page
Subsection	2.4.2 The Compromise Decision Support Problem	31
Section	2.5 Summary of Literature Reviewed and Requirements Identified	32
Subsection	2.5.1 Requirements Identified from the Literature Review	40
Section	2.6 The Structural Translation Gap	41
Section	2.7 Limitations of Existing Approaches	43
Subsection	2.7.1 Limitations of the Systems Archetype Literature	43
Subsection	2.7.2 Limitations of the System Dynamics Literature	43
Subsection	2.7.3 Limitations of the cDSP Literature	44
Subsection	2.7.4 Limitations of the Rural CPSS Literature	44
Section	2.8 Research Positioning	44
Section	2.9 What Has Been Presented in Chapter 2 and What Is Next	46
Chapter	CHAPTER 3: CONTEXT-AWARE CONTROL SYSTEM ARCHITECTURE AND METHOD	48
Section	3.1 Introduction	48
Section	3.2 Dilemma Triangle Method as the Foundation of the Framework	50
Subsection	3.2.1 Role of the Dilemma Triangle Method	50
Subsection	3.2.2 Local Knowledge	51
Subsection	3.2.3 Uncertainty	52
Subsection	3.2.4 Limited Resources	53
Section	3.3 Identification of Structural Tensions	53
Section	3.4 Core Layer: Unified Satisficing Architecture	56
Subsection	3.4.1 Purpose of the Core Layer	56
Subsection	3.4.2 Satisficing Engine	57
Subsection	3.4.3 Closed-Loop Operational Evaluation	58
Subsection	3.4.4 Reuse Heuristics	59
Section	3.5 Semi-Core Layer: Structural Translation and Constraint Mapping	60
Subsection	3.5.1 Purpose of the Semi-Core Layer	60
Subsection	3.5.2 Construction of Causal Loop Diagrams (Method Step 2)	61
Subsection	3.5.3 Archetype Interpretation (Method Step 3)	63

Level	Heading	Page
Subsection	3.5.4 Archetype Quantification (Method Step 4 - Translation)	65
Subsection	3.5.5 Constraint Mapping and cDSP Translation (Method Step 5)	67
Section	3.6 Application Layer: Execution and Policy Evaluation	68
Subsection	3.6.1 Sense: Establishing the Operating Context	70
Subsection	3.6.2 Frame	71
Subsection	3.6.3 Decide	72
Subsection	3.6.4 Act	72
Subsection	3.6.5 Learn	73
Section	3.7 Operational Workflow of the Framework	74
Chapter	CHAPTER 4: TEST PROBLEMS FOR THE CONTEXT-AWARE CONTROL SYSTEM FRAMEWORK	77
Unnumbered	Applying the Sense, Frame, Decide, Act, and Learn Workflow Across the Three Test Problems	78
Section	4.1 Photovoltaic Battery Microgrid under the Fixes That Fail Archetype	82
Subsection	4.1.1 Summary of the Fixes That Fail Translation	83
Subsection	4.1.2 Why This Test Problem Is Appropriate for Validation	83
Subsection	4.1.3 Problem Context	84
Subsection	4.1.4 CLD Construction (Method Step 2)	85
Subsection	4.1.5 Archetype Interpretation (Method Step 3)	86
Subsection	4.1.6 Archetype Quantification and cDSP Formulation (Method Steps 4 and 5)	87
Subsection	4.1.7 Results and Interpretation	91
Section	4.2 Rural Water Allocation under the Tragedy of the Commons Archetype	97
Subsection	4.2.1 Problem Context	98
Subsection	4.2.2 CLD Construction (Method Step 2)	100
Subsection	4.2.3 Archetype Interpretation (Method Step 3)	103
Subsection	4.2.4 Translation to cDSP (Method Steps 4 and 5)	104
Subsection	4.2.5 Results and Interpretation	111
Section	4.3 Rural Healthcare Triage under the Shifting the Burden Archetype	122
Subsection	4.3.1 Problem Statement	122

Level	Heading	Page
Subsection	4.3.2 CLD Construction (Method Step 2)	123
Subsection	4.3.3 Archetype Interpretation (Method Step 3)	124
Subsection	4.3.4 Compromise Decision Support Problem Formulation	124
Subsection	4.3.5 Results and Interpretation	127
Section	4.4 Verification and Validation	134
Subsection	4.4.1 Structure of the Validation Process	135
Subsection	4.4.2 Empirical Structural Validity (Quadrant 2)	136
Subsection	4.4.3 Empirical Performance Validity (Quadrant 3)	137
Subsection	4.4.4 Summary of Validation Evidence	139
Section	4.5 What Has Been Presented in Chapter 4 and What Is Next	140
Chapter	CHAPTER 5: CLOSING REMARKS	142
Section	5.1 Design Functionalities and Development Outcomes	142
Subsection	5.1.1 F1: Problem Framing Functionality	142
Subsection	5.1.2 F2: Translation Functionality	143
Subsection	5.1.3 F3: Satisficing Decision Functionality	144
Subsection	5.1.4 F4: Closed-Loop Execution Functionality	144
Subsection	5.1.5 F5: Cross-Domain Demonstration Functionality	145
Subsection	5.1.6 Synthesis Across the Three Test Problems	145
Subsection	5.1.7 Assumptions Underlying the Framework and Its Demonstration	146
Section	5.2 Limitations of the Framework	147
Section	5.3 Future Research Directions	149
Section	5.4 Closing Statement	150
Unnumbered	REFERENCES	152

LIST OF TABLES

Table	Caption	Page
Table 1.1	Connection between design functionalities and chapters.	2
Table 1.2	Design functionalities, rationale, method elements, test problem evidence, and development outcomes.	13
Table 1.3	Mapping of design functionalities, method elements, and development outcomes.	19
Table 1.4	One-page traceability sheet: Design Functionality, Method Element, CACCS Layer, Test Problem, Validation Quadrant, Development Outcome.	22
Table 2.1	Literature review of rural CPSS, systems archetypes, and cDSP satisficing.	32
Table 4.1	CLD-to-cDSP mapping for the photovoltaic battery microgrid.	87
Table 4.2	Baseline policy results across seven weight scenarios over the 30-day evaluation horizon.	91
Table 4.3	CACCS policy results (discharge ceiling $B = 50$ kWh/day) across seven weight scenarios over the 30-day evaluation period.	93
Table 4.4	Comparison of baseline and CACCS policy outcomes across the seven weight scenarios.	96
Table 4.5	Mapping of CLD Variables to cDSP Formulation Elements (Water Allocation).	106
Table 4.6	Summary of 30-day outcomes across three water allocation scenarios.	111
Table 4.7	Day-by-day water allocation results.	112
Table 4.8	Validation Square status for the CACCS framework.	139

LIST OF FIGURES

Figure	Caption	Page
Figure 3.1	Dilemma Triangle Method: Local Knowledge, Uncertainty, and Limited Resources.	51
Figure 3.2	Context-Aware Control System architecture and method workflow.	55
Figure 3.3	Three-layer Context-Aware Control System architecture.	56
Figure 4.1	Causal loop diagram for the photovoltaic battery microgrid under the Fixes That Fail archetype.	86
Figure 4.2	Ternary plots illustrating baseline policy energy allocation among critical services.	93
Figure 4.3	Ternary plots showing CACCS policy outcomes for energy allocation among critical services.	95
Figure 4.4	Causal loop diagram representing the Tragedy of the Commons structure.	102
Figure 4.5	Ternary plots for TP2 Baseline (agg=1.30).	118
Figure 4.6	Ternary plots for TP2 CACCS Dom Priority (wD=0.60, wA=0.30, wE=0.10).	119
Figure 4.7	Ternary plots for TP2 CACCS Agri+Ent Priority (wD=0.20, wA=0.40, wE=0.40).	119
Figure 4.8	TP2 superimposed satisficing space.	120
Figure 4.9	Causal loop diagram representing the Shifting the Burden structure.	124
Figure 4.10	Ternary plots for TP3 Baseline.	130
Figure 4.11	Ternary plots for TP3 CACCS SP (wS=0.60, wP=0.25, wM=0.15).	131
Figure 4.12	Ternary plots for TP3 CACCS BA (wS=0.20, wP=0.55, wM=0.25).	131
Figure 4.13	TP3 superimposed satisficing space.	131

ABSTRACT

In this thesis, the author develops a Context-Aware Control System (CACCS) framework for the adaptive and satisficing design of rural cyber-physical-social systems (rural CPSS). Rural service systems, including groundwater allocation networks, off-grid microgrids, and primary healthcare clinics, operate under uncertainty, limited resources, and locally embedded practices that conventional optimization approaches cannot reliably represent. The author addresses the structural translation gap in this thesis: the absence of a replicable method for translating the structural components of systems archetypes into operational decision formulations.

In the CACCS framework, the author integrates the Dilemma Triangle Method, causal loop diagram construction, systems archetype interpretation, an archetype quantification method, and the compromise Decision Support Problem within a three-layer architecture comprising a Core Layer, a Semi-Core Layer, and an Application Layer. Decision-makers use archetype quantification to translate balancing loops into service-target goal equations, reinforcing loops into degradation penalties and feasibility constraints, structural tensions into weighted trade-offs, and delayed feedback effects into repeated closed-loop operational constraints.

The author demonstrates the framework through three rural test problems, each governed by a distinct systems archetype: a photovoltaic battery microgrid under the Fixes That Fail archetype, a groundwater allocation system under the Tragedy of the Commons archetype, and a primary healthcare clinic under the Shifting the Burden archetype. For each test problem, the author constructs a causal loop diagram, interprets it through its governing archetype, translates the structure into

a compromise Decision Support Problem, and evaluates satisficing policies against archetype-unaware baseline policies through closed-loop execution across a thirty-day decision horizon.

Across the three test problems, designers observe that decision makers use CACCS satisficing policies to preserve structural state variables, shift shortfalls in directions consistent with the governing weight structure and avoid the structural failure modes that researchers associate with archetype-unaware baseline policies. In this thesis, the author develops a problem-framing method for rural cyber-physical-social systems. The author shows how decision-makers recognize the structural tension in a rural service problem before they formulate a satisficing operational decision. Decision-makers use archetype quantification to translate that frame into decision variables, constraints, deviation terms, weights, and state updates. Decision-makers then use the compromise Decision Support Problem to formulate satisficing actions that account for uncertainty, limited resources, local knowledge, and stakeholder priorities.

GLOSSARY

CACCS	Context-Aware Control System, the layered architecture developed in this thesis (Core Layer, Semi-Core Layer, Application Layer).
cDSP	Compromise Decision Support Problem, a structured method for formulating multi-goal engineering decisions under constraints (Mistree, Hughes, and Bras, 1993), organized as Given, Find, Satisfy, and Minimize.
CLD	Causal loop diagram, a diagram that represents a system's structure as variables connected by causal links organized into balancing and reinforcing loops.
CPSS	Cyber-physical-social system, an engineered system in which technical infrastructure, human behavior, and social practice interact as coupled structural components.
DTM	Dilemma Triangle Method, the problem-formulation method used to define the rural problem space through the drivers of local knowledge, uncertainty, and limited resources.
rural CPSS	A rural service system in which technical infrastructure, environmental conditions, and human and community behavior interact as coupled structural components rather than as separable technical and social layers.
Archetype quantification	The method, developed in Section 3.5.4, for translating the structural components of a systems archetype (balancing loops, reinforcing loops, structural tensions, delayed effects) into cDSP elements (goal equations, constraints, weighted trade-offs, state updates).

- Balancing loop** A feedback loop that stabilizes or constrains system behavior, denoted B in causal loop diagrams.
- Reinforcing loop** A feedback loop that amplifies system behavior, denoted R in causal loop diagrams.
- Satisficing** A decision philosophy, introduced by Simon (1956), that seeks solutions which are good enough relative to an aspiration level rather than solutions that are globally optimal.
- Structural translation gap** The absence, prior to this thesis, of a replicable method for translating the structural components of systems archetypes into operational decision formulations.
- TP1, TP2, TP3** Shorthand references to Test Problem 1 (photovoltaic battery microgrid), Test Problem 2 (rural water allocation), and Test Problem 3 (healthcare triage), respectively.

CHAPTER 1

FRAME OF REFERENCE: PROBLEM FRAMING

BEFORE SATISFICING DECISION MAKING IN

RURAL CYBER-PHYSICAL-SOCIAL SYSTEMS

The principal goal in this thesis is to establish approaches that are suitable for the adaptive and satisficing design of a Context-Aware Control System for rural cyber-physical-social systems (rural CPSS). Designers define a rural CPSS as a rural service system in which technical infrastructure, environmental conditions, and human and community behavior interact as coupled structural components rather than as separable technical and social layers. In this system, designers manage uncertainty and limited resources, integrate local knowledge, identify structural tensions, and translate archetypal feedback structures into operational decision formulations at design time. The principal motivation in this thesis is to generate a repeatable method for translating archetypal feedback structures into compromise Decision Support Problem formulations for rural cyber-physical-social systems, resulting in a layered Context-Aware Control System architecture that is adaptive, modular, and satisficing across rural conditions.

In this chapter, a foundation is laid for achieving the goal addressed in this thesis. The problem context, problem statement, research gap, research objective, and rationale for a satisficing philosophy are presented in Sections 1.1 through 1.5. The Dilemma Triangle Method based problem space formulation, including drivers, focus statements, tension matrix, and dilemmas, is presented in Section 1.6. The author summarizes the thesis contribution in Section 1.7. Design functionalities, the linkage between design functionalities and development outcomes, located research gaps, and the mapping of design functionalities to method elements and development outcomes are presented in Sections 1.8 through 1.11. Frameworks

considered in this thesis are: (1) Dilemma Triangle Method, (2) causal loop diagrams and systems archetypes, and (3) compromise Decision Support Problem.

In Chapter 1, research gaps are identified and based on the identified gaps, five design functionalities for the method are specified that are addressed throughout the thesis. In Table 1.1, the author presents the connection between design functionalities and chapters.

Table 1.1. Connection between design functionalities and chapters. F = foundation. J = justification. Q1-Q4 = quadrants of the Validation Square.

Design Functionalities	Ch. 1	Ch. 2	Ch. 3	Ch. 4	Ch. 5
F1. Decision-makers need to recognize the structural tension in a rural cyber-physical-social situation before they formulate the decision problem.	F	J	Q1	Q2, Q3	Q4
F2. Decision-makers need to translate a qualitative frame, expressed through causal loop diagrams and systems archetypes, into cDSP decision variables, constraints, deviation terms, weights, and state updates.	F	J	Q1	Q2, Q3	Q4
F3. Decision-makers need to formulate satisficing operational decisions that account for uncertainty, limited resources, local knowledge, and stakeholder priorities.	F	J	Q1	Q2, Q3	Q4
F4. Decision-makers need to update state variables and reformulate decisions as rural operating conditions change.	F	J	Q1	Q2, Q3	Q4

F5. Designers need to show that decision-makers can use the same method structure for energy, water, and healthcare test problems, without redesigning the method for each domain.					
---	--	--	--	--	--

1.1 Problem Context

Rural communities across the developing and developed world depend on service systems, electricity microgrids, water supply networks, primary healthcare clinics, that must operate reliably under conditions substantially different from the urban, industrial, and defense contexts for which most control and decision-support methods were originally developed. In rural cyber-physical-social systems, physical infrastructure is often intermittent or aging, environmental conditions vary unpredictably across seasons and years, and the humans who operate these systems, a microgrid operator managing a battery bank, a water system manager balancing competing user demands, a clinic administrator allocating scarce staff hours, must make repeated operational decisions with incomplete sensing, limited connectivity, and heavy reliance on locally embedded knowledge rather than centralized data infrastructure.

These operators are rarely provided with formal decision-support tools, and where such tools do exist, they are typically imported from urban or industrial contexts and assume conditions, continuous sensing, stable infrastructure, a single well-defined objective, that do not hold in rural environments. As a result, rural decision-makers most often rely on heuristics developed through experience: deliver more energy when a shortage is reported, allocate more water when a request is loudest, treat the patient in front of you before considering the clinic's preventive care backlog. These heuristics are not irrational; they reflect a reasonable response to an immediately visible symptom. What they frequently lack is any structured

mechanism for recognizing that the immediately visible symptom may be only the surface expression of a deeper, recurring structural pattern, a pattern that systems thinking research has long described through devices such as causal loop diagrams and systems archetypes, but that has not, prior to this thesis, been connected to an operational decision-support method a rural practitioner could actually use.

The rural cyber-physical-social systems examined in this thesis, an off-grid photovoltaic battery microgrid, a shared-aquifer water allocation system, and a primary healthcare triage system, are chosen because each exhibits this pattern concretely. In each case, an operator facing an immediate service shortfall has available a corrective action that provides fast, visible relief. In each case, that same corrective action interacts with a slower-moving structural process, battery degradation, aquifer depletion, erosion of preventive health capability, that is not visible on the time scale of a single decision but that determines whether the system remains viable over weeks and months. This is the operating context within which the present thesis is situated: rural cyber-physical-social systems in which decision-makers face structurally patterned problems without a structured way to recognize the pattern before acting on it. In this thesis, the author represents battery degradation, aquifer depletion, and health capability erosion using the specific relationships in each test problem's compromise Decision Support Problem, as described in Chapter 4. The author simplifies the relationships that represent these processes because real physical and social processes are more complex and, in several cases, nonlinear. In Section 5.2, the author returns to this scope decision as a limitation of the framework

1.2 Problem Statement

People who design conventional control systems for rural infrastructure typically assume that the problem confronting the decision-maker has already been correctly framed. Framing, in the sense used throughout this thesis, means recognizing which structural tension is actually in play in a given operating situation, whether a shared resource is being depleted through the uncoordinated action of many independent users, whether a short-term corrective action is quietly eroding the long-term capacity that action depends on, or whether an immediate symptomatic response is displacing investment in a slower, more fundamental solution, before any decision about what to do is made. Conventional control system design typically bypasses this recognition step entirely. The designer takes the decision problem as given, specifies an objective function or a set of rules, and proceeds directly to optimizing or automating a decision within whatever frame is implicitly assumed, usually the frame of an ordinary, single-period resource-allocation problem in which more input produces more output and there is no meaningful difference between a good short-term decision and a good long-term one.

In rural cyber-physical-social systems, this assumption is frequently unwarranted, and the consequences of the mismatch are not merely theoretical. A microgrid operator who frames declining service quality as simply a matter of insufficient daily energy delivery, rather than recognizing that aggressive battery discharge is itself accelerating the degradation that produces future shortfalls, will select a policy that appears to solve the problem today while making the underlying problem worse. A water system manager who frames rising household shortfalls as simply a matter of insufficient allocation, rather than recognizing that every stakeholder group's individually reasonable withdrawal is collectively depleting a shared aquifer, will select a policy that increases extraction precisely when restraint is most needed. A clinic administrator who frames a growing backlog of acute cases

as simply a triage problem, rather than recognizing that chronic under-investment in preventive care is what is driving the growth in acute presentations, will allocate ever more capacity to symptomatic treatment while the community's underlying health capability continues to decline. In each case, the decision-maker who misframes the situation does not fail to solve a problem; they solve the wrong problem, competently and consistently, for as long as the underlying structural tension remains unrecognized.

In this thesis, the author addresses how a rural cyber-physical-social control system can support the act of framing itself, not merely the decision that follows it. Designers ask how such a system can be structured so that a decision-maker, or a system acting on a decision-maker's behalf, can sense the current operating context, frame the situation by recognizing the structural tension it embodies, make a satisficing decision consistent with that framing, act on the decision, and learn from the outcome as the situation evolves, all within the resource, connectivity, and information constraints that characterize rural environments. Recognizing the situation correctly, before deciding what to do about it, is the central and, designers argue, largely unaddressed problem this thesis takes up (Schön, 1983; Rittel and Webber, 1973)

1.3 Research Gap

The literature reviewed in Chapter 2 establishes that three distinct, but related, gaps exist in the bodies of work most relevant to this thesis. Systems archetypes, established by Senge (1990) and extended by Braun (2002) and Kim (1994), provide a rich qualitative vocabulary for recognizing recurring structural patterns, Fixes That Fail, Tragedy of the Commons, Shifting the Burden, Limits to Growth, but the literature stops at recognition and description; no existing method specifies how the structural components of an identified archetype, its balancing loops, reinforcing loops, structural tensions, and delayed effects, become the decision

variables, constraints, and objectives of an operational decision model. The compromise Decision Support Problem, introduced by Mistree, Hughes, and Bras (1993) and applied widely across engineering design and, more recently, rural energy contexts (Bertus et al., 2012), provides a mature and flexible satisficing decision framework, but no existing cDSP application uses a systems archetype as the structural basis for its formulation; existing formulations are built from domain-specific engineering models rather than from a prior act of structural framing. Rural cyber-physical-social systems, meanwhile, remain substantially understudied relative to the urban, industrial, and smart-grid contexts that dominate the broader CPSS literature (Sobb, Turnbull, and Moustafa, 2023), and no existing rural CPSS framework functions across more than one service domain at a time.

Underlying all three gaps is a more fundamental one that this thesis foregrounds and that the existing literature does not name directly: every reviewed body of work, whether concerned with archetypes, satisficing decision formulations, or rural systems specifically, implicitly assumes that the decision-maker has already correctly framed the situation before the method in question is applied. Researchers have not developed, within any of the three reviewed bodies of literature, a structured way of supporting that framing step itself, of moving a decision-maker from an unexamined presenting symptom to an explicit recognition of the structural tension generating that symptom, before any decision model is built. Section 1.10 develops each of the three gaps in detail and Section 2.6 names and defines the resulting structural translation gap precisely.

1.4 Research Objective

The objective of this thesis is to develop and demonstrate a method by which a rural cyber-physical-social system can support a decision-maker in framing the structural tension underlying an operational situation, recognizing whether the situation is governed by a pattern such as Fixes That Fail, Tragedy of the Commons, or Shifting

the Burden, and in carrying that framing through, without loss of structural consistency, into a satisficing operational decision. Designers pursue this objective without requiring continuous connectivity, autonomous machine learning, or computationally intensive infrastructure, since none of these can be assumed to be reliably available in the rural environments the author addresses in this thesis.

Designers pursue the objective specifically by integrating four elements within a single architecture: the Dilemma Triangle Method, which supports the initial act of framing a rural problem space; systems archetype interpretation, which characterizes the structural tension the framing reveals; an archetype quantification method, developed as part of this thesis, which translates that qualitative structural understanding into quantitative decision constructs; and the compromise Decision Support Problem, which computes a satisficing operational decision consistent with the framing. Designers demonstrate this integrated method across three rural test problems, an off-grid photovoltaic battery microgrid, a shared-aquifer water allocation system, and a rural healthcare triage system, chosen because each is governed by a distinct, well-documented systems archetype and situated in a distinct rural service domain.

1.5 Rationale for Satisficing

Framing a problem correctly does not, by itself, produce a single optimal decision. What framing supplies is a recognition of which structural tension is in play and, consequently, which competing goals must be balanced against one another; it does not supply a ranking that reduces those competing goals to a single number to be maximized or minimized. Once a rural water allocation problem is correctly framed as a Tragedy of the Commons, for example, the appropriate response is not to find the mathematically optimal extraction rate under an assumed single objective, since no such single objective exists that adequately represents the community's actual priorities. The appropriate response is a compromise among the water needs of

domestic users, agricultural users, and enterprise users, a compromise whose specific balance depends on locally determined priorities that the framing step identifies as being in tension but does not, and should not, resolve on the analyst's behalf.

For this reason, designers adopt Simon's (1956) satisficing philosophy, operationalized through the compromise Decision Support Problem of Mistree, Hughes, and Bras (1993), as the decision-making mechanism that acts once a situation has been framed. Simon's foundational insight, that decision-makers operating under bounded rationality, incomplete information, and limited computational capacity cannot realistically find globally optimal solutions and instead seek decisions that are good enough relative to an aspiration level, is particularly apt for rural decision environments, where information is incomplete, feedback is delayed, and operating conditions shift unpredictably across operational cycles. Guo, Allen, and Mistree (2024) provide a formal engineering argument reinforcing this choice, showing that in nonlinear, nonconvex design problems with multiple conflicting objectives, satisficing solutions that meet necessary Karush-Kuhn-Tucker conditions achieve better practical performance than optimization approaches that impose idealized assumptions the problem does not actually satisfy.

Designers adopt satisficing, and specifically the cDSP construct, because it is structurally suited to acting on a frame: it accommodates multiple, partially conflicting goals; it makes trade-offs among those goals explicit through weighted deviation terms rather than hiding them inside a single aggregated objective; and it allows a decision-maker's priorities to be represented directly as weights, rather than presupposing that framing alone can supply the single correct answer that optimization requires.

1.6 DTM-Based Problem Space Formulation

Decision-makers use the Dilemma Triangle Method as the specific mechanism through which framing happens within this thesis, and it is deliberately positioned prior to, and separate from, the decision formulation that follows it. Before any decision model is constructed, designers use the DTM to identify the three structural drivers that recur across rural problem spaces: local knowledge, the accumulated, often informal experience and practice through which rural communities have historically managed their own resources; uncertainty, the unpredictable variation in environmental conditions, infrastructure reliability, and information availability that rural decision-makers cannot eliminate through better forecasting alone; and limited resources, the constrained budgets, weak infrastructure, and restricted technical manpower that bound what any proposed intervention can realistically require. Designers then examine the tensions that arise among these three drivers, oral transfer of knowledge against missing sensor data, trust gaps against environmental shocks, adaptation requirements against limited technical manpower, and from these tensions' designers derive the specific operational dilemmas that characterize a given rural problem space.

The author deliberately keeps this formulation separate from the decision formulation that follows it later in the method. The Dilemma Triangle Method frames the situation: it establishes which drivers and tensions are present and, through the causal loop diagram and archetype interpretation steps that follow it, which structural pattern those tensions produce. The compromise Decision Support Problem, introduced later in Chapter 3 through the archetype quantification method, decides within that frame: it does not itself determine which archetype is present or which tensions matter, and it is not invoked until the framing step has already established what kind of situation the decision-maker is in. Keeping these two steps distinct, framing first, deciding second, is what allows the same

downstream decision machinery, the same Core Layer satisficing engine, and the same Semi-Core Layer translation procedure, to be reused unchanged across the three structurally different rural test problems examined in Chapter 4. Section 3.2 develops the Dilemma Triangle Method in full, including its application to each of the three test problems.

1.7 Thesis Contributions

In this thesis, the author develops a problem-framing method for rural cyber-physical-social systems. A decision-maker uses this method to recognize the structural tension in a rural service problem before formulating a satisficing operational decision. A decision-maker uses systems archetypes to name the structural tension. A decision-maker uses archetype quantification to translate that named tension into operational decision constructs. A decision-maker then uses the compromise Decision Support Problem to formulate a decision that accounts for uncertainty, limited resources, and local knowledge. Taken individually, each contribution, an architecture, a translation method, an execution procedure, could be described as a technical artifact in its own right. Taken together, and this is the point made explicit at the outset rather than left for the reader to infer from the individual chapters, the three contributions constitute something more specific than a collection of technical artifacts: a replicable way for a decision-maker to move from an unexamined rural operating situation to a structurally sound, satisficing operational decision, without losing, at any step along the way, the connection between how the situation was framed and what was ultimately decided.

The principal contribution of this thesis, stated at this level of generality, is a method for cognitive framing in context-aware control under uncertainty: a replicable procedure through which a decision-maker recognizes the structural tension governing a rural cyber-physical-social system before committing to an

operational decision, and carries that recognition through into a satisficing control action. Decision-makers use the Dilemma Triangle Method for this initial framing step. Systems archetype interpretation (Senge, 1990; Braun, 2002) characterizes the structural tension the framing reveals. Decision-makers use archetype quantification to translate that qualitative structural understanding into quantitative decision constructs. The compromise Decision Support Problem (Mistree, Hughes, and Bras, 1993) computes a satisficing action consistent with the framing.

The Dilemma Triangle Method, archetype interpretation, archetype quantification, and the compromise Decision Support Problem are not, individually, the contribution of this thesis, in the sense that none of them is newly invented here in isolation from the others. They are the specific, largely pre-existing mechanisms that a designer assembles, in a particular order and through a particular translation procedure that is new, to make cognitive framing operational within a closed-loop, context-aware control architecture. The author identifies the contribution as the assembly and the translation procedure and decision-makers gain the ability to frame a rural problem correctly as a functionality of the method.

1.8 Design Functionalities

Each of the five design functionalities below states what decision-makers or designers need to be able to do with the method, and points back to the central spark of this thesis, that people need ways to frame problems before they make decisions in rural cyber-physical-social systems.

F1. Problem framing functionality

Decision-makers need to recognize the structural tension in a rural cyber-physical-social situation before they formulate the decision problem.

F2. Translation functionality

Decision-makers need to translate a qualitative frame, expressed through causal loop diagrams and systems archetypes, into cDSP decision variables, constraints, deviation terms, weights, and state updates.

F3. Satisficing decision functionality

Decision-makers need to formulate satisficing operational decisions that account for uncertainty, limited resources, local knowledge, and stakeholder priorities.

F4. Closed-loop execution functionality

Decision-makers need to update state variables and reformulate decisions as rural operating conditions change.

F5. Cross-domain demonstration functionality

Designers need to show that decision-makers can use the same method structure for energy, water, and healthcare test problems, without redesigning the method for each domain.

1.9 Design Functionalities to Development Outcomes

Each design functionality connects to a development outcome that decision-makers or designers can obtain by using the method. In Table 1.2, the author presents this connection explicitly. The Design Functionality column states what decision-makers or designers need to do. The Why Decision Makers Need This Functionality column states the rationale linking the functionality to the method. In the Method Element column, the author states which part of the method decision-makers or designers use to satisfy the functionality.

Table 1.2. Design functionalities, rationale, method elements, test problem evidence, and development outcomes

Design Functionality	Why Decision Makers Need This Functionality	Method Element Used by Designers or Decision Makers	Test Problem Evidence	Development Outcome
F1. Problem framing functionality	Decision-makers who do not recognize the structural tension in a rural situation risk solving the wrong problem: an ordinary allocation view rather than the archetype actually governing the situation.	Dilemma Triangle Method, causal loop diagram construction, and systems archetype interpretation (Sections 3.2, 3.5.2, 3.5.3)	Sections 4.1.2, 4.2.1, 4.3.1: each test problem opens with the visible symptom, the risk of misframing it, and the named archetype	Decision-makers use the method to move from a visible service symptom to a named structural frame before they formulate the decision problem
F2. Translation functionality	A named frame is only useful if	Archetype quantification method and	Sections 4.1.6, 4.2.4, 4.3.2: CLD-	Decision-makers use archetype

	decision-makers can convert it into decision variables, constraints, deviation terms, weights, and state updates that DSIDES can act on.	CLD-to-cDSP translation (Section 3.5.4, 3.5.5)	to-cDSP mapping tables for each test problem	quantification to translate a named frame into operational decision constructs
F3. Satisficing decision functionality	Decision-makers facing multiple, partially conflicting goals under uncertainty need a decision formulation that accounts for stakeholder priorities rather than a	Compromise Decision Support Problem, Core Layer satisficing engine (Section 3.4)	Sections 4.1.6, 4.2.4, 4.3.2: Given-Find-Satisfy-Minimize formulations for each test problem	Decision-makers use the compromise Decision Support Problem to make satisficing decisions within the frame

	single optimal answer.			
F4. Closed-loop execution functionality	Rural operating conditions change from one decision cycle to the next, so decision-makers need a way to update state variables and reformulate the decision rather than apply a fixed rule.	Application Layer, closed-loop state update procedure (Sections 3.6, 3.4.3)	Sections 4.1.7, 4.2.5, 4.3.3: 30-day closed-loop trajectories for each test problem	Decision-makers use repeated state updates to examine future operating conditions after each decision
F5. Cross-domain demonstration functionality	A method that only works for one rural service domain provides limited evidence that	The same eight-step workflow (Section 3.7) applied without modification	Chapter 4 overall: energy (4.1), water (4.2), and healthcare (4.3) test problems	Designers apply the same method structure to three rural service domains

	designers can reuse it elsewhere without redesigning it.	to three test problems	using one unmodified method structure	
--	--	------------------------	---------------------------------------	--

The relationship among these columns traces the logical spine of the thesis. In Chapter 2, the author justifies the design functionalities by locating the gap in the existing literature. In Chapter 3, the author describes the method elements that satisfy each functionality. In Chapter 4, the author demonstrates each functionality through three test problems. In Chapter 5, the author states the resulting development outcomes.

1.10 Research Gaps

Three distinct but related gaps exist in the literature. The author identifies these gaps through the critical literature review of Chapter 2 and are summarized here in order to connect each gap directly to the corresponding design functionality and method element of Section 1.11.

Gap 1: The Literature Reviewed Does Not Specify How to Translate Systems Archetype Structures into Operational Decision Formulations

Systems archetypes have been established as powerful diagnostic tools in the work of Senge (1990), Braun (2002), and Kim (1994). Researchers use archetypes to identify recurring structural failure patterns and to suggest qualitative interventions. However, the literature reviewed here does not specify how the structural components of an archetype become operational decision constructs. Among the works reviewed, none states how a balancing relief loop becomes a service-target

goal equation, how a reinforcing erosion loop becomes a degradation penalty or feasibility constraint, how a structural tension becomes a weighted trade-off in a satisficing objective, or how a delayed feedback effect becomes a repeated closed-loop operational constraint. As far as the reviewed literature indicates, this translation gap leaves decision-makers without a way to realize the practical utility of archetype-based reasoning for operational decision support. In response to this gap, F2, the translation functionality, is specified. The archetype quantification method introduced in Chapter 3 and demonstrated in Chapter 4 addresses this gap directly.

Gap 2: No Reviewed Framework Integrates Satisficing Decision Formulations with Rural CPSS Structural Reasoning

The compromise Decision Support Problem has been applied across engineering design, materials selection, supply chain management, and rural energy contexts. By reading the literature reviewed in Chapter 2, one understands that the cDSP is well suited for managing multi-goal trade-offs under realistic constraints. However, none of the cDSP applications reviewed uses systems archetype structures as the basis for formulation; the reviewed applications proceed from domain-specific engineering models rather than from structural feedback diagnosis. Based on this review, existing cDSP formulations do not preserve structural consistency with the feedback patterns that govern recurring operational failures in rural systems. In response to this gap, F1, the problem framing functionality, is specified. The three-layer Context-Aware Control System architecture introduced in Chapter 3 addresses this gap directly.

Gap 3: Rural CPSS Remains Understudied in the Reviewed Systems Design Literature

By reading Sobb et al. (2023), one understands that rural CPSS remains substantially understudied, with manufacturing, smart cities, and smart grid as the dominant application areas. The rural service system models reviewed address single domains, and none of them provides a generalizable operational framework that functions across energy, water, and healthcare domains simultaneously. In response to this gap, F4, the closed-loop execution functionality, and F5, the cross-domain demonstration functionality, are specified. The closed-loop execution procedure of the Application Layer, demonstrated across three distinct rural domains in Chapter 4, addresses this gap directly. These three gaps are not independent. Based on the literature reviewed, the absence of a structural translation method prevents the integration of systems thinking with decision formulations, and neither reviewed body of work has been applied to the rural CPSS context that most urgently requires operational decision support. The author addresses all three gaps within a unified framework in this thesis. F3, the satisficing decision functionality, is not a response to a fourth gap in this sense: the compromise Decision Support Problem is an established, well-validated decision formulation in its own right, and Section 1.5 states the rationale for adopting it rather than a gap it fills. F3 draws on this established method component; F1, F2, F4, and F5 respond directly to the three gaps above.

1.11 Design Functionalities, Method Elements, and Development Outcomes

In Table 1.3, the author presents the complete mapping of design functionalities, method elements, and development outcomes. The author uses this table as the traceable spine of the thesis argument. Every section of Chapter 3 connects to a design functionality through this table. Every test problem of Chapter 4 provides evidence that decision-makers or designers satisfy a functionality. Every

development outcome of Chapter 5 follows from the development outcome column of Table 1.2.

Table 1.3. Mapping of design functionalities, method elements, and development outcomes.

	Design Functionality	Method Element	Development Outcome
F1	Decision-makers need to recognize the structural tension in a rural cyber-physical-social situation before they formulate the decision problem.	Dilemma Triangle Method and systems archetype interpretation enable decision-makers to name the structural tension governing a rural situation. Addressed in Chapter 3, Sections 3.2 and 3.5.3.	Decision-makers move from a visible service symptom to a named structural frame before they formulate the decision problem.
F2	Decision-makers need to translate a qualitative frame, expressed through causal loop diagrams and systems archetypes, into cDSP decision variables, constraints, deviation terms, weights, and state updates.	Archetype quantification and CLD-to-cDSP translation give decision-makers a replicable procedure for this conversion. Addressed in Chapter 3, Sections 3.5.4 and 3.5.5.	Decision-makers use archetype quantification to translate a named frame into operational decision constructs.

F3	Decision-makers need to formulate satisficing operational decisions that account for uncertainty, limited resources, local knowledge, and stakeholder priorities.	The compromise Decision Support Problem and Core Layer satisficing engine give decision-makers a decision formulation suited to multiple, partially conflicting goals. Addressed in Chapter 3, Section 3.4.	Decision-makers use the compromise Decision Support Problem to make satisficing decisions within the frame.
F4	Decision-makers need to update state variables and reformulate decisions as rural operating conditions change.	Decision-makers use the Application Layer's closed-loop state update procedure to carry structural memory across decision cycles. Addressed in Chapter 3, Sections 3.6 and 3.4.3.	Decision-makers use repeated state updates to examine future operating conditions after each decision.
F5	Designers need to show that decision-makers can use the same method structure for energy, water, and healthcare test problems, without redesigning the method for each domain.	The eight-step workflow of Section 3.7, applied without modification, gives designers a basis for comparing method use across domains. Addressed in Chapter 4.	Designers apply the same method structure to three rural service domains.

The five design functionalities are not independent. F1 gives decision-makers the framing step. F2 gives decision-makers the translation step that populates the frame with domain-specific decision constructs. F3 gives decision-makers the satisficing decision step. F4 gives decision-makers the closed-loop update step. F5 gives designers the cross-domain evidence that the same method structure applies without redesign. Together, these five functionalities give designers and decision-makers a complete, integrated, and replicable method for framing rural cyber-physical-social problems before deciding what to do about them.

Designers use the Validation Square structure of Pedersen, Emblemssvag, Bailey, Allen, and Mistree (2000) to organize evidence that the method satisfies the five design functionalities. Theoretical Structural Validity (Quadrant 1) is addressed in Chapter 3 through the internal consistency of the three-layer architecture. Empirical Structural Validity (Quadrant 2) is addressed in Chapter 4 through the selection and justification of three structurally distinct test problems. Empirical Performance Validity (Quadrant 3) is addressed in Chapter 4 through evidence that decision-makers can frame, translate, formulate, execute, and update decisions across three rural service domains. Theoretical Performance Validity (Quadrant 4) is addressed in Chapter 5 through the argument that using the same method structure across three domains supports its use in domains not yet tested. In Table 1.4, the author consolidates these relationships into a one-page traceability sheet.

Table 1.4. One-page traceability sheet: Design Functionality, Method Element, CACCS Layer, Test Problem, Validation Quadrant, Development Outcome.

Design Functionality	Method Element	CACCS Layer	Test Problem	Validation Quadrant	Development Outcome
---------------------------------	---------------------------	------------------------	-------------------------	--------------------------------	--------------------------------

F1: problem framing functionality	Dilemma Triangle Method, CLD construction, archetype interpretation	Semi- Core Layer	All 3 (TP1, TP2, TP3): each opens with a visible symptom, a risk of misframing, and a named archetype	Q1 (Theoretical Structural) + Q2 (Empirical Structural)	Decision- makers move from a visible symptom to a named structural frame before formulating the decision problem
F2: translation functionality	Archetype quantification, CLD- to-cDSP translation	Semi- Core Layer	TP1 (Fixes That Fail), TP2 (Tragedy of the Commons) , TP3 (Shifting the Burden)	Q2 (Empirical Structural) + Q3 (Empirical Performance)	Decision- makers use archetype quantification to translate a named frame into operational decision constructs
F3: satisficing	Compromise Decision	Core Layer	All 3: Given-	Q2 (Empirical)	Decision- makers use

decision functionalit y	Support Problem, Core Layer satisficing engine		Find- Satisfy- Minimize formulatio ns for each test problem	Structural) + Q3 (Empirical Performan ce)	the cDSP to make satisficing decisions within the frame
F4: closed- loop execution functionalit y	Closed- loop state update procedure	Applicati on Layer	All 3: 30- day closed- loop trajectorie s for each test problem	Q3 (Empirical Performan ce)	Decision- makers use repeated state updates to examine future operating conditions after each decision
F5: cross- domain demonstrati on functionalit y	Eight-step workflow applied without modificatio n	All three layers	Chapter 4 overall: energy, water, and healthcare test problems using one unmodifie	Q4 (Theoretic al Performan ce)	Designers apply the same method structure to three rural service domains

			d method structure		
--	--	--	-----------------------	--	--

CHAPTER 2

LITERATURE REVIEW: RURAL CYBER- PHYSICAL-SOCIAL SYSTEMS, SYSTEMS ARCHETYPES, AND SATISFICING DECISION FORMULATIONS

2.1 Scope of the Review

Rural cyber-physical-social systems present decision-making challenges that differ fundamentally from those addressed in conventional engineering control systems. In rural environments, infrastructure is intermittent, feedback is delayed, operating conditions shift unpredictably, and locally embedded practices interact with technical systems in ways that standard optimization frameworks cannot easily represent. Despite growing recognition of these challenges, the literature has not yet produced a unified operational framework that integrates structural feedback reasoning, systems archetype interpretation, and satisficing decision formulations within a context-aware control architecture for rural service systems.

In this chapter, designers review three bodies of literature that together define the research gap that designers address. In Section 2.2, designers examine the literature on rural cyber-physical-social systems, focusing on how researchers have characterized the structural tensions and recurring failure patterns that appear across rural energy, water, and healthcare domains. In Section 2.3, designers review the systems archetypes and systems dynamics literature, focusing on how recurring feedback structures have been identified and interpreted qualitatively. In Section 2.4, designers review the satisficing decision formulation literature, focusing on the compromise Decision Support Problem construct and its applications in multi-goal engineering design contexts. In Section 2.5, designers consolidate the review through a structured literature table and identify requirements that existing work

has not yet fulfilled. In Section 2.6, designers state the gap in the literature that motivates the present thesis.

Table 2.1 summarizes the literature reviewed in this chapter. Designers organize the reviewed work by domain in Table 2.1 and identify which features - rural context, domain coverage, feedback structure, archetype interpretation, quantitative translation, cDSP formulation, closed-loop evaluation, and multi-domain applicability - each work addresses. The requirements identified from this review are listed at the end of the chapter.

2.2 Rural Cyber-Physical-Social Systems

The concept of cyber-physical-social systems emerged from the recognition that modern engineered systems cannot be understood through their technical components alone. Ning and Liu (2015) established the foundational CPSS paradigm by arguing that effective system design must account for interactions among the cyber space, the physical space, and the social space simultaneously. This framing distinguishes CPSS from conventional cyber-physical systems by explicitly incorporating human behavior, community practices, and social dynamics as structural components of the system rather than external disturbances.

2.2.1 The CPSS Paradigm and Its Rural Extension

Subsequent work extended the CPSS framework in several directions. Yin et al. (2020) proposed a 7C data-driven CPSS framework for social manufacturing, demonstrating that the social dimension introduces qualitatively different modeling requirements than purely technical system descriptions. Sobb et al. (2023) conducted a holistic review of the CPSS literature from 2021 to 2023 and identified manufacturing, smart cities, and smart grid applications as the dominant areas of research activity. Critically, their review found that rural CPSS - service systems characterized by resource scarcity, infrastructure instability, limited connectivity,

and strong dependence on locally embedded practices - represent a substantially understudied subclass within the broader CPSS literature.

The gap that Sobb, Turnbull, and Moustafa (2023) identify is consequential. Rural CPSS such as groundwater allocation systems, off-grid electricity microgrids, and primary healthcare clinics share structural features that differ from the urban and industrial systems that dominate the CPSS literature. These features include: stochastic resource availability governed by seasonal and environmental cycles; operational decision-making under incomplete information and delayed feedback; locally developed practices and heuristics that are embedded in community behavior rather than encoded in formal system specifications; and recurring structural tensions that generate persistent failure patterns rather than isolated technical failures. None of these features are well addressed by the data-driven, optimization-oriented frameworks that characterize the existing CPSS literature.

2.2.2 Resource Dynamics and Feedback in Rural Service Systems

The rural service literature has documented recurring resource dynamics across multiple domains. In the water domain, Libey et al. (2022) developed a system dynamics model of rural water supply and maintenance in East Africa using causal loop diagrams to characterize the feedback relationships between pump usage, borehole functionality, and maintenance financing. Their model identified reinforcing loops that amplify aquifer depletion under increasing demand and balancing loops that constrain service continuity through equipment degradation. While this work successfully models the structural dynamics of rural water systems, it does not translate the identified feedback structures into operational decision formulations or evaluate policy performance through closed-loop comparison.

In the energy domain, Bertus et al. (2012) applied the Dilemma Triangle Method and the compromise Decision Support Problem to rural microgrid power allocation for quality-of-life improvement in rural India. Their work demonstrated that the

cDSP can be used to manage competing energy loads across water, safety, education, and social activity categories in resource-constrained rural environments. However, the formulation did not incorporate systems archetype interpretation, and the analysis was limited to a single decision period without closed-loop operational evaluation across multiple domains.

In the water-energy-food nexus literature, several studies have applied system dynamics models to simulate resource interactions across coupled rural sectors. These studies consistently identify feedback structures that produce depletion, tipping point behavior, and delayed degradation effects under uncoordinated extraction. However, the literature on WEF nexus dynamics does not translate these structural insights into operational decision formulations that a rural decision-maker could execute under real operating constraints.

Across these domains, the rural service literature has produced detailed descriptions of the feedback structures that drive recurring failure patterns but has not developed a generalizable method for translating those structures into operational decision models that preserve structural consistency, function under uncertainty, and remain implementable under limited resource conditions.

2.3 Systems Archetypes and Structural Feedback Interpretation

Systems archetypes were introduced by Senge (1990) as recurring structural patterns in organizational and social systems that produce characteristic behavior over time regardless of the specific domain in which they appear. Senge identified archetypes including Fixes That Fail, Tragedy of the Commons, Shifting the Burden, and Limits to Growth as templates for diagnosing structural dysfunction and identifying high-leverage intervention points. Braun (2002) extended this work by providing detailed structural documentation of eight core archetypes, describing the interacting balancing and reinforcing loops that define each archetype's behavior and the characteristic management errors they reveal.

2.3.1 Origins and Qualitative Formulation

The systems thinking literature established archetypes as powerful qualitative diagnostic tools. Kim (1994) provided archetype templates for organizational diagnosis, and subsequent work applied archetypes across domains ranging from healthcare policy to environmental management to organizational strategy. The consistent finding across this literature is that archetype-based diagnosis enables decision-makers to trace recurrent operational failures back to structural causes rather than attributing them to isolated events or individual decisions.

However, this body of work is almost entirely qualitative. Researchers describe archetypes through causal loop diagrams and narrative explanations and state intervention strategies at a conceptual level. Researchers have not provided a method for translating the structural components of an archetype into quantitative decision variables, operational constraints, and satisficing objectives.

2.3.2 System Dynamics and Quantitative Simulation

System dynamics, introduced by Forrester (1961), provides a quantitative complement to the qualitative archetype framework. Through stock and flow diagrams, causal loop diagrams, and differential equation formulations, system dynamics enables designers to simulate the behavior of complex feedback structures over time. The method has been applied broadly across business, transportation, public policy, environmental management, and engineering design.

Mandegari, Bhalerao, Allen and Mistree (2024) integrated system dynamics with the cDSP to predict robust satisficing solutions for evolving system states. Their method used system dynamics to simulate system behavior at multiple past and present states and applied the cDSP to identify weight ranges that produce satisficing solutions robust across multiple states. This work represents the most direct precedent for the present thesis and demonstrates the structural compatibility between system dynamics modeling and cDSP satisficing. However, their test

problem used a coal combustion power plant scenario rather than a rural CPSS context, and the formulation did not use systems archetypes as the structural basis for the cDSP.

Bhalerao, Honeycutt, Allen and Mistree (2023) applied evidentiary and interpretative analysis to frame wicked problems through causal loop diagrams and systems thinking, demonstrating that structural diagnosis through feedback analysis can inform decision formulation. Bhalerao, Honeycutt, Allen, and Mistree established the qualitative-to-quantitative translation pathway that designers extend through archetype quantification.

2.3.3 The Gap Between Archetype Identification and Decision Formulation

The central gap in the systems archetype and system dynamics literature is the absence of a structured, replicable method for translating identified archetype structures into operational decision constructs. Existing work either remains at the qualitative archetype level without quantification or applies system dynamics simulation without systematic connection to a satisficing decision framework. No existing method specifies how the balancing relief loops of an archetype become operational service objectives, how reinforcing erosion loops become degradation penalties or feasibility constraints, how structural tensions become weighted trade-offs, or how delayed feedback effects become repeated closed-loop operational constraints. In this thesis, the author develops this translation as the core methodological contribution.

2.4 Satisficing Decision Formulations and the Compromise Decision Support Problem

The satisficing philosophy was introduced by Simon (1956) as a behavioral theory of decision-making under bounded rationality. Simon argued that decision-makers operating with incomplete information, limited computational capacity, and uncertain futures cannot find globally optimal solutions and instead seek decisions

that are good enough relative to an aspiration level. This observation is particularly relevant for rural decision environments where information is incomplete, feedback is delayed, and operating conditions change unpredictably across operational cycles.

2.4.1 The Satisficing Philosophy

Guo, Allen and Mistree (2024) provided a formal engineering argument for satisficing over optimization in model-based design contexts. Their analysis demonstrated that in nonlinear, nonconvex design problems with multiple conflicting objectives, satisficing solutions that meet necessary Karush-Kuhn-Tucker conditions achieve better practical performance than optimization approaches that impose idealized assumptions about the problem structure. This argument motivates the adoption of a satisficing philosophy within the present thesis, where rural operational problems consistently violate the assumptions required for optimization to produce reliable results.

2.4.2 The Compromise Decision Support Problem

The compromise Decision Support Problem (cDSP) construct was introduced by Mistree, Hughes and Bras (1993) as a structured method for formulating and solving engineering design problems involving multiple conflicting goals under constraints. The cDSP organizes decision problems through four components: Given conditions that describe the problem context, Find variables that the decision-maker controls, Satisfy constraints that must be met, and Minimize an Archimedean deviation function that aggregates weighted shortfalls from goal targets. The construct is implemented through the DSIDES computational environment documented by Sharma, Allen and Mistree (2023).

Subsequent applications have extended the cDSP across engineering design, materials selection, supply chain management, and socio-technical systems. Sani

Mohammed et al. (2023) applied the cDSP to military supply chain resilience, demonstrating that the construct can handle multi-goal operational problems under disruption. Bertus et al. (2012) applied the cDSP to rural microgrid power allocation, establishing its applicability in resource-constrained rural environments. The present thesis extends these applications by connecting archetype identification directly to cDSP formulation and demonstrating closed-loop satisficing policy evaluation across three distinct rural domains.

Across these applications, the cDSP has consistently demonstrated its ability to produce satisficing solutions that balance competing operational priorities under realistic constraints. However, none of the existing applications has systematically connected archetype identification to cDSP formulation through a replicable translation method, demonstrated closed-loop operational evaluation across multiple rural domains under distinct archetypes, or extended the framework to water and healthcare domains where the Tragedy of the Commons and Shifting the Burden archetypes govern structural behavior.

2.5 Summary of Literature Reviewed and Requirements Identified

In Table 2.1, designers summarize the literature reviewed in this chapter. Rows represent individual papers. Columns represent features relevant to the research gap designers address in this thesis. An asterisk (*) indicates that the corresponding paper addresses the feature.

Table 2.1. Literature review of rural CPSS, systems archetypes, and cDSP satisficing.

Paper	Rural Context	Energy Domain	Water Domain	Health Domain	CLD / Feedback	Systems Archetype	Archetype Quantified	System Dynamics	eDSP / Satisficing	Closed- Loop	Uncertainty Handled	Operational Decision	Multi- Domain	Objective / Method	Gap Addressed
Senge (1990) The Fifth Discipline					*	*								Qualitative introduction of systems archetypes as recurring structural patterns in organizations	Archetypes remain qualitative; no translation to quantitative decision formulations
Braun (2002) The System Archetypes					*	*								Documents eight core archetypes including Fixes That Fail, Tragedy of the Commons, Shifting the	Diagnostic only; no method to translate archetypal structure into operational models

													Burden, Limits to Growth	
Forrester (1961) Industrial Dynamics				*			*						Establishes system dynamics as a method for modeling feedback structures and simulating complex system behavior	Simulation without satisficing decision construct; no operational policy evaluation
Mistree, Hughes & Bras (1993) cDSP Construct								*		*	*		Introduces the compromise Decision Support Problem for multi-goal satisficing in engineering design	Applied in engineering design contexts; not extended to rural service systems or archetype-driven formulations
Simon (1956)								*		*			Establishes	Theoretical

Ratio nal Choic e													satisficin g as a bounded- rationalit y decision strategy for complex problems with incomple te informati on	foundation ; no operational formatio n for multi- stakeholde r rural systems
Guo, Allen & Mistr ee (2024) Opti mize or Satisfi ce								*		*	*		Demonst rates that satisficin g solutions meeting necessary KKT condition s outperfor m optimizat ion in nonlinear , nonconve x	Engineerin g design focus; not applied to rural service systems or feedback- driven archetypes

													engineeri ng design	
Mand egari, Bhale rao, Allen & Mistr ee (2024) Predi cting Futur e States				*			*	*	*	*	*		Integrate s system dynamics and cDSP to predict robust satisficin g solutions for evolving system states using ternary satisficin g spaces	Single- domain test problem; archetypes not used as the structural basis for cDSP formulatio n
Bhale rao, Hone ycutt, Allen & Mistr ee (2023) Frami ng Wicke d				*	*					*			Uses evidentia ry and interpreta tive analysis to frame wicked problems through causal loop diagrams	Qualitative framing only; no translation to decision constructs or policy evaluation

Problems														and systems thinking	
Bertus, Khosrojerdi, Panchal, Allen & Mistricee (2012) Dilemmas in Rural India	*	*							*		*	*		Applies DTM and cDSP to microgrid power allocation for rural quality of life improvement	Single-domain energy focus; no archetype interpretation or closed-loop operational evaluation
Ning & Liu (2015) Cyber-Physical-Social Systems													*	Defines CPSS paradigm integrating cyber, physical, and social spaces for context-aware	Urban and industrial focus; rural-specific constraints and local knowledge not addressed

													system behavior	
Yin et al. (2020) Data- Drive n CPSS										*		*	Proposes 7C framewor k for data- driven CPSS in social manufact uring; extends CPS to include social dimensio ns	Manufactu ring context; no rural service application or satisficing decision logic
Sobb et al. (2023) Holist ic Revie w of CPSS										*		*	Systemat ic review of CPSS literature; identifies manufact uring, smart cities, and smart grid as dominant applicati on areas	Rural CPSS identified as understudi ed; no operational framework s for resource- constraine d rural environme nts

Libey et al. (2022) Rural Water System Dynamics (ASC E J. Env. Eng.)	*		*		*			*		*			System dynamics model of rural water supply and maintenance in East Africa using CLDs and feedback loop analysis	Descriptive simulation only; no decision formulation, archetype identification, or policy comparison
Sharma, Allen & Mistricee (2023) Concurrently Coupled Systems								*	*	*	*		Applies cDSP to concurrently and hierarchically coupled engineering systems through DSIDES	Coupled systems in engineering design; no rural CPSS context or archetype-driven formulation
Sani Moha								*		*	*		Uses cDSP	Supply chain

- R1. The framework must address rural cyber-physical-social systems characterized by uncertainty, limited resources, and locally embedded practices that conventional optimization approaches cannot reliably represent.
- R2. Designers require the framework to identify recurring structural feedback patterns through causal loop diagrams and to interpret them through systems archetypes in order to diagnose the structural causes of operational failure.
- R3. Designers require the framework to provide a replicable method for translating the structural components of identified archetypes, balancing loops, reinforcing loops, structural tensions, and delayed effects, into decision variables, operational constraints, deviation terms, and weighted satisficing objectives.
- R4. Designers require the framework to use the compromise Decision Support Problem construct to formulate operational decision problems that balance competing service goals under realistic resource and infrastructure constraints.
- R5. Designers require the framework to evaluate satisficing policies through closed-loop operational execution across multiple decision periods and compare CACCS policies against archetype-unaware baseline policies.
- R6. Designers require the framework to demonstrate domain independence by applying the same translation method across at least three distinct rural service domains each governed by a distinct systems archetype.

2.6 The Structural Translation Gap

Through the literature review in Sections 2.2 through 2.4, designers identify a gap that designers name precisely: the structural translation gap. This gap refers to the absence of a structured, replicable method for translating the structural components

of identified systems archetypes into operational decision constructs that a rural decision-maker can execute under real operating constraints.

The structural translation gap has four specific dimensions. First, no existing method specifies how a balancing relief loop in a causal loop diagram becomes a service-target goal equation in a compromise Decision Support Problem. The relationship between a balancing loop corrective action variable and the corresponding deviation variable in a cDSP goal equation is implicit in all existing work: published applications choose a goal equation and a decision variable because they are the natural quantities for the domain, not because a documented procedure derived the goal equation from an identified balancing loop, so the same corrective-action-to-deviation-variable mapping cannot be reproduced by a practitioner working from a CLD alone. Second, no existing method specifies how a reinforcing erosion loop becomes a degradation penalty or feasibility constraint. Reinforcing loops are recognized as sources of long-term structural deterioration in the archetype literature, but no existing work encodes the reinforcing loop state-update behavior as a closed-loop operational constraint in a decision formulation. Third, no existing method specifies how the structural tension between competing loops, which is the defining feature of every archetype, becomes a weighted trade-off in a satisficing objective. The weight structure of the Archimedean deviation function in the cDSP is the natural representation of this tension, but no existing work makes this connection explicit. Fourth, no existing method specifies how delayed feedback effects in an archetype become repeated closed-loop operational constraints across multiple decision cycles.

These four dimensions together define the methodological contribution of this thesis. The archetype quantification method of Section 3.5.4 and the CLD-to-cDSP mapping tables of Sections 4.1 through 4.3 address each dimension explicitly and systematically.

2.7 Limitations of Existing Approaches

The literature reviewed in this chapter has produced valuable but partial contributions. Understanding their limitations is necessary in order to understand what designers contribute and why.

2.7.1 Limitations of the Systems Archetype Literature

The primary limitation of the systems archetype literature is that it operates entirely at the qualitative level. Senge (1990) and Braun (2002) provide rich structural descriptions of recurring feedback patterns, but neither provides a procedure for translating archetype structure into quantitative decision models. Kim (1994) provides archetype templates for organizational diagnosis, but the interventions recommended are stated in conceptual terms. The gap between qualitative archetype interpretation and operational decision support remains open in all existing archetype work. A secondary limitation is that existing archetype work is designed primarily for organizational diagnosis rather than for operational control. The goal of archetype-based diagnosis is to help managers understand why organizations behave in counterproductive ways. The present thesis pursues a different goal: to produce operational decision policies that respond to archetype-identified structural tensions in real time.

2.7.2 Limitations of the System Dynamics Literature

The primary limitation of the system dynamics literature, in the context of this thesis, is that simulation is not the same as decision support. By reading Mandegari et al. (2024), one understands that system dynamics and the cDSP are structurally compatible and can be combined to identify robust satisficing solutions for evolving system states. However, their method uses system dynamics primarily as a demand-forecasting tool for the cDSP rather than as a structural basis for formulating the cDSP goal equations and constraints. The archetype structure of the test problem is

not identified, and the connection between feedback loops and cDSP elements is not made explicit.

2.7.3 Limitations of the cDSP Literature

By reading the work of Mistree, Hughes and Bras (1993), Sharma et al. (2023), and Sani Mohammed et al. (2023), one understands that the cDSP is a powerful and flexible construct for multi-goal satisficing under constraints. However, existing cDSP formulations are derived from domain-specific engineering models. The goal equations, constraints, and deviation terms are chosen based on what matters in the domain, not based on the structural feedback patterns that produce recurring operational failures. The consequence is that existing cDSP applications cannot distinguish between a problem that has a Fixes That Fail structure and a problem that has a Tragedy of the Commons structure.

2.7.4 Limitations of the Rural CPSS Literature

By reading Sobb et al. (2023), one understands that rural CPSS is understudied. By reading Libey et al. (2022) and Bertus et al. (2012), one understands that feedback modeling and decision support have been applied in rural domains separately but not integrated within a unified operational framework. No existing rural CPSS paper provides a method that a practitioner could apply to a new rural domain by following a replicable set of steps.

2.8 Research Positioning

In this thesis, the author positions the research at the intersection of four existing bodies of work: systems archetype theory, system dynamics modeling, the compromise Decision Support Problem, and rural cyber-physical-social systems. No prior researcher has occupied this intersection.

By reading the systems archetype literature, one understands the structural patterns that govern recurring rural operational failures but finds no translation to operational decision models. By reading the system dynamics literature, one finds simulation tools that model feedback behavior but no connection to satisficing decision formulations. By reading the cDSP literature, one finds powerful multi-goal satisficing constructs but no connection to archetype-based structural reasoning. By reading the rural CPSS literature, one finds domain-specific models but no generalizable framework. The present thesis fills this intersection by providing the structural translation layer that none of these bodies of work currently provides. In Table 2.1 of this chapter, designers show that no existing paper addresses all of the features required for this translation. The requirements R1 through R6 of Section 2.5.1 state what a framework that closes the structural translation gap must do. The framework developed in Chapter 3 and demonstrated in Chapter 4 fulfills all six requirements.

In this chapter, designers identify a specific and consequential gap in the existing literature. Researchers have developed qualitative tools for identifying recurring structural feedback patterns in complex systems but have not produced a method for translating those patterns into operational decision formulations. Researchers have developed satisficing decision frameworks capable of managing multi-goal trade-offs under constraints but have not connected those frameworks to the structural feedback reasoning that diagnoses why rural service systems fail repeatedly. Researchers have characterized the structural complexity of rural CPSS but have not demonstrated a generalizable operational framework that preserves structural consistency between feedback diagnosis, decision formulation, and policy evaluation.

In this thesis, designers address this gap by developing a Context-Aware Control System framework that integrates Dilemma Triangle Method-based problem

formulation, causal loop diagram construction, systems archetype interpretation, archetype quantification, and compromise Decision Support Problem formulation within a unified closed-loop operational architecture. Designers demonstrate the framework through three rural test problems: a photovoltaic battery microgrid under the Fixes That Fail archetype, a groundwater allocation system under the Tragedy of the Commons archetype, and a primary healthcare clinic under the Shifting the Burden archetype, each exercising a distinct archetype across a distinct rural service domain. Through this demonstration, designers show that decision-makers obtain structurally consistent, operationally meaningful, and archetype-aware satisficing policies that outperform archetype-unaware baseline policies across all three domains.

2.9 What Has Been Presented in Chapter 2 and What Is Next

In this chapter, designers reviewed the literature in three topic areas relevant to the research gap addressed in this thesis. In Section 2.2, designers reviewed the literature on rural cyber-physical-social systems. By reading this literature, one understands that rural CPSS is characterized by resource scarcity, infrastructure instability, and strong dependence on locally embedded practices, and that researchers have not yet addressed these characteristics within a generalizable operational decision method, based on the literature reviewed here.

In Section 2.3, designers reviewed the literature on systems archetypes and system dynamics. By reading this literature, one understands that recurring structural feedback patterns have been identified and documented qualitatively, and that researchers have not yet translated these patterns into operational decision formulations, based on the literature reviewed here.

In Section 2.4, designers reviewed the literature on satisficing decision-making and the compromise Decision Support Problem. By reading this literature, one understands that the cDSP is a powerful construct for multi-goal satisficing under constraints, and that no existing application connects the cDSP to archetype-based structural reasoning in a rural CPSS context.

In Sections 2.5 through 2.8, designers consolidated the literature into a structured table, identified six requirements for the framework, named and defined the structural translation gap, identified the limitations of each body of reviewed literature, and positioned the present research at the intersection of the four reviewed bodies of work.

In Chapter 3, designers present the Context-Aware Control System architecture and method that addresses the structural translation gap. The three-layer architecture, the archetype quantification method, and the closed-loop execution procedure together constitute the research contribution through which designers fill the gap designers identify in this chapter.

CHAPTER 3

Context-Aware Control System Architecture and Method

3.1 Introduction

In this thesis, designers develop a Context-Aware Control System (CACCS) framework for rural cyber-physical-social systems operating under uncertainty, incomplete information, limited resources, and evolving environmental conditions. Designers integrate systems thinking, dilemma-based problem formulation, archetype interpretation, and satisficing decision-making into a unified operational framework that supports adaptive rural decision-making across multiple domains. Designers do not treat rural systems as isolated engineering systems with stable operating conditions. Instead, designers examine them as cyber-physical-social systems in which infrastructure, human behavior, environmental variability, and operational constraints continuously interact. In such environments, operational failures rarely emerge from a single technical malfunction. Repeated failures instead arise from structural tensions embedded within feedback relationships among resources, infrastructure, stakeholders, and environmental conditions. In many rural environments, decision-makers operate with incomplete data, weak infrastructure, intermittent connectivity, delayed feedback, and locally embedded practices that cannot easily be represented through rigid technical rules. Conventional optimization approaches frequently assume stable operating conditions, continuous information flow, and clearly defined objectives. However, these assumptions rarely remain valid in rural contexts. A mathematically optimal solution may become operationally infeasible if it depends upon infrastructure, communication systems, or technical support that does not exist within the operating environment.

For this reason, designers adopt a satisficing philosophy throughout the framework. Rather than searching for globally optimal decisions, designers seek operationally acceptable decisions that preserve service continuity, maintain structural feasibility, and remain implementable under active constraints. Within the framework, decision-making therefore emerges through trade-offs among uncertainty, degradation, stakeholder priorities, resource limitations, and long-term system preservation. Designers organize the framework through three interacting architectural layers: Core Layer, Semi-Core Layer, Application Layer

In this thesis, Frame is the activity through which a decision-maker identifies the structural tension in a rural cyber-physical-social situation before formulating the decision problem. Designers use the architecture to determine how structural reasoning is represented within the framework, and use the method to determine how rural operational problems move from qualitative interpretation toward executable satisficing decisions. The complete workflow of the framework proceeds through several stages. First, designers use the Dilemma Triangle Method to define the rural problem space through interacting drivers and tensions. Second, designers construct causal loop diagrams in order to identify dominant feedback relationships governing operational behavior. Third, designers interpret the causal structures through systems archetypes in order to identify recurring structural tensions. Fourth, decision-makers translate the interpreted structural behavior into compromise Decision Support Problem formulations. Finally, designers evaluate satisficing policies through repeated closed-loop operational execution across multiple rural scenarios and archetype combinations. Through this process, designers operationalize systems thinking within a context-aware control architecture for rural cyber-physical-social systems.

Designers demonstrate this method through three rural test problems presented in Chapter 4. In Section 4.1, designers examine a photovoltaic battery microgrid under

the Fixes That Fail archetype. In Section 4.2, designers examine a rural water allocation problem under the Tragedy of the Commons archetype. In Section 4.3, designers examine a rural healthcare triage problem under the Shifting the Burden archetype. Designers refer to these test problems throughout the present chapter when designers illustrate how each method step applies to a specific rural environment.

3.2 Dilemma Triangle Method as the Foundation of the Framework

The Dilemma Triangle Method (DTM) serves as the foundation for defining the rural problem space within the Context-Aware Control System. Rather than beginning with technical variables or optimization objectives, the framework begins by identifying the dominant tensions that shape rural decision environments.

3.2.1 Role of the Dilemma Triangle Method

Designers use the Dilemma Triangle Method because rural systems cannot be fully understood through isolated technical failures alone. In many cases, operational difficulties emerge from interactions among environmental uncertainty, resource limitations, and locally embedded practices. These interacting tensions produce recurring operational dilemmas that persist even when short-term technical interventions are applied. The DTM therefore provides a structured mechanism for identifying the underlying forces that repeatedly generate system instability.

Within the framework, designers define the rural problem space through three primary drivers:

- Local Knowledge
- Uncertainty
- Limited Resources

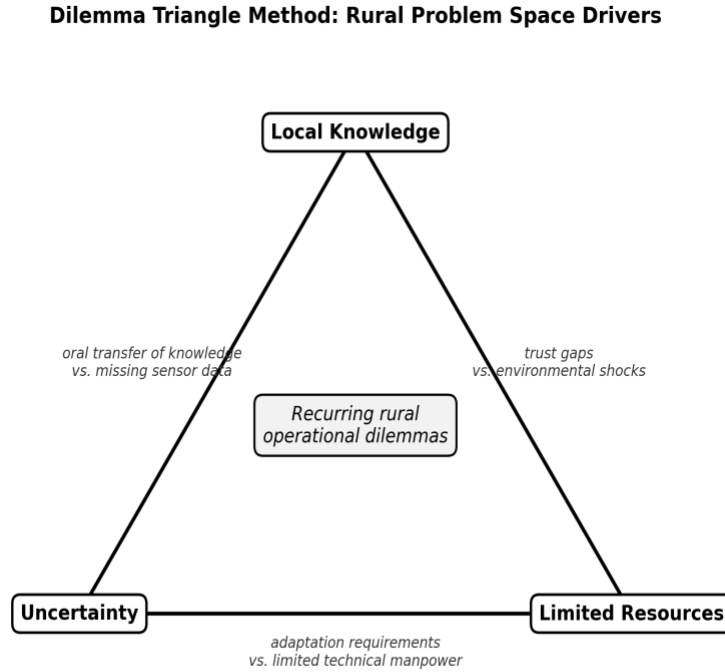


Figure 3.1. Dilemma Triangle Method: Local Knowledge, Uncertainty, and Limited Resources as the primary drivers of the rural problem space.

In Figure 3.1, the author represents these three drivers as the vertices of the Dilemma Triangle. These drivers represent recurring structural conditions that appear across multiple rural domains, including the energy, water, and healthcare domains examined in the test problems of Chapter 4.

3.2.2 Local Knowledge

Local knowledge represents the accumulated experience, practices, and decision heuristics developed within rural communities over long periods of time. This knowledge often includes agricultural methods, water management practices, informal healthcare responses, and operational workarounds developed under resource constraints. In many rural environments, this knowledge remains largely informal and context specific. It is transmitted through practice, observation, and

lived experience rather than through standardized technical documentation. Although such knowledge frequently supports operational resilience, it also creates challenges for formalized control systems because it cannot easily be represented through rigid technical rules or centralized optimization structures. Within the framework, local knowledge therefore represents both:

- a valuable operational resource,
- and a source of structural complexity.

Designers require the framework to preserve the ability to incorporate local practices while simultaneously maintaining consistent operational reasoning.

3.2.3 Uncertainty

Uncertainty represents one of the defining characteristics of rural cyber-physical-social-systems. Environmental conditions shift unpredictably, infrastructure may fail intermittently, and causes of operational disruptions often remain ambiguous. Unlike highly controlled industrial systems, rural environments frequently operate with incomplete information and delayed feedback. Weather patterns, resource availability, social responses, and operational conditions may change rapidly without reliable forecasting mechanisms. Decision-makers therefore cannot assume that future operating conditions will remain stable. Within the Context-Aware Control System, uncertainty is not treated as an exceptional disturbance. Instead, uncertainty is treated as a permanent operating condition that influences all stages of the decision process.

The framework therefore prioritizes:

- operational flexibility,
- satisficing trade-offs,
- and structurally feasible responses

rather than rigid optimization under fixed assumptions.

3.2.4 Limited Resources

Limited resources represent the third foundational driver within the Dilemma Triangle Method. Rural systems frequently operate under:

- constrained budgets,
- weak infrastructure,
- low technical manpower,
- intermittent connectivity,
- and restricted maintenance capability.

These limitations influence both the design and execution of operational decisions. A theoretically effective control strategy may still become infeasible if it depends upon infrastructure, expertise, or communication systems that are unavailable within the operating environment. For this reason, the framework emphasizes:

- modularity,
- lightweight operational structures,
- and feasible satisficing policies rather than computationally intensive or infrastructure-dependent approaches.

The limited-resource driver therefore ensures that the framework remains grounded in realistic operational constraints.

3.3 Identification of Structural Tensions

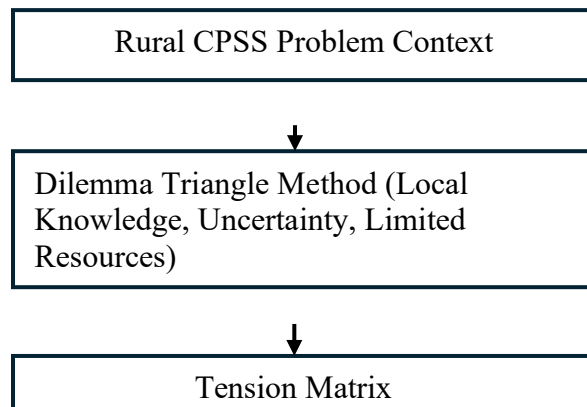
After defining the primary drivers, designers organize the associated issues into a tension matrix in order to identify recurring structural conflicts within the rural problem space. The purpose of the tension matrix is not simple categorization. Instead, designers use the matrix to identify interacting conditions that repeatedly

generate instability across multiple rural domains. Examples of these tensions include:

- oral transfer of knowledge versus missing sensor data,
- trust gaps versus environmental shocks,
- and adaptation requirements versus limited technical manpower.

Designers do not interpret these tensions as temporary operational failures. Instead, designers interpret them as persistent structural relationships that continuously influence system behavior across repeated operational cycles. From these tensions, designers derive recurring dilemmas that shape the operational logic of the framework. For example, when sensor infrastructure fails simultaneously with loss of local operational knowledge, decision-makers lose both technical and experiential information sources. Similarly, when environmental shocks intensify under limited manpower conditions, operational adaptation becomes increasingly difficult to sustain. These dilemmas later guide:

- causal loop construction,
- archetype interpretation,
- and cDSP formulation



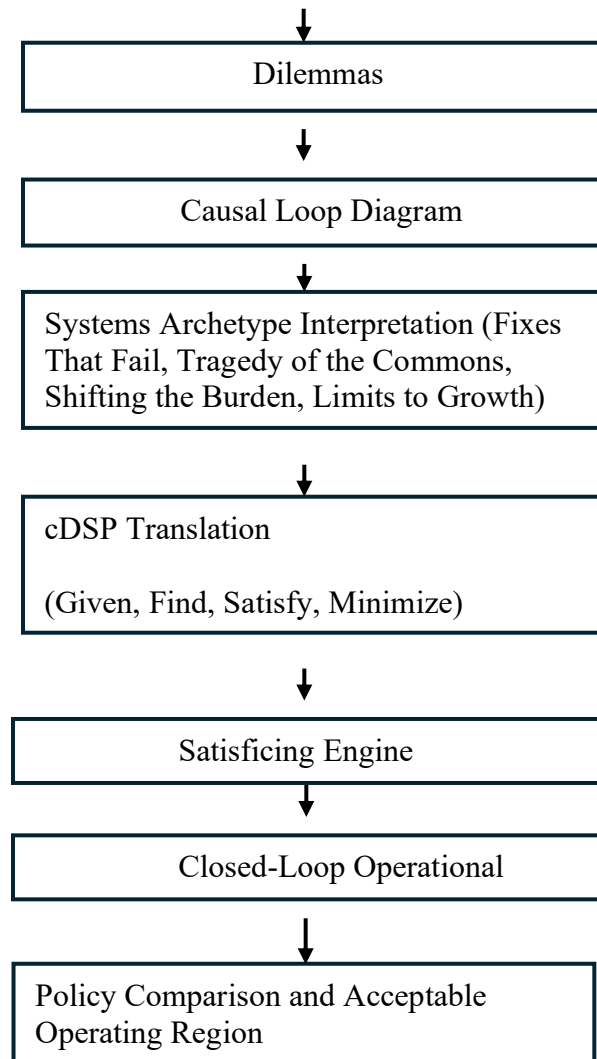


Figure 3.2. Context-Aware Control System method workflow.

In Figure 3.2, the author shows the complete operational workflow of the framework. The workflow begins with DTM-based problem formulation, moves through causal loop and archetype interpretation, translates structural tensions into a cDSP formulation, and evaluates satisficing policies through repeated closed-loop operational evaluation. Designers describe each of these workflow stages in

Sections 3.4 through 3.6 and consolidate them as an end-to-end procedure in Section 3.7. Designers then apply this workflow to three rural test problems in Chapter 4.

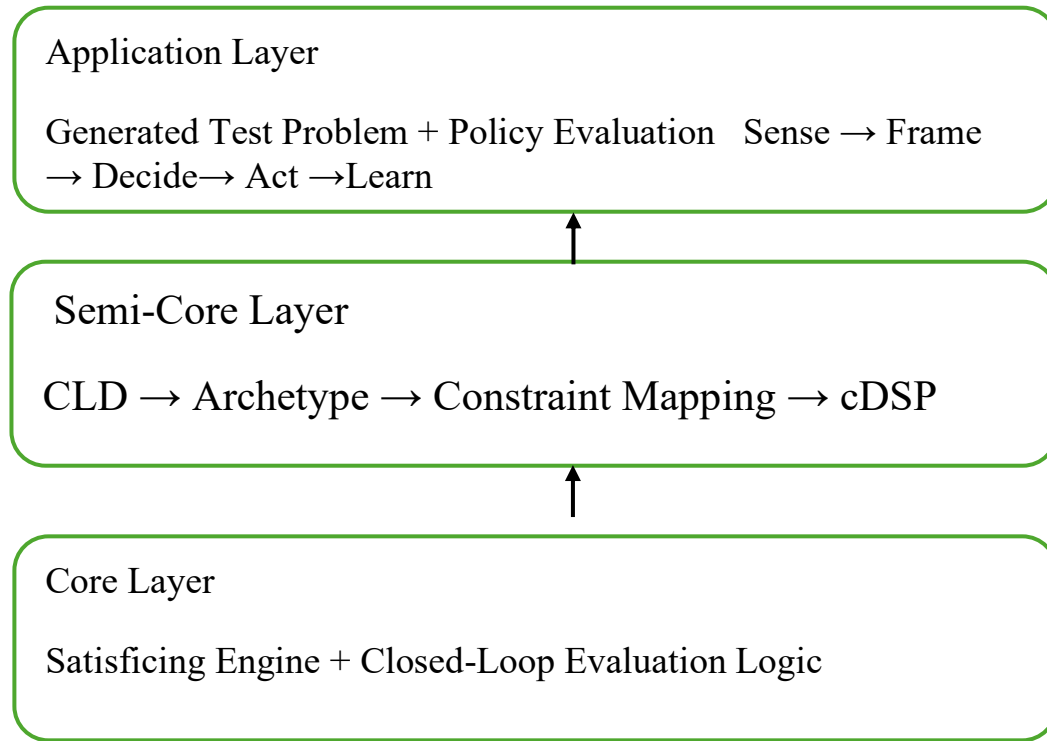


Figure 3.3. Three-layer Context-Aware Control System architecture.

In Figure 3.3, the author shows the three architectural layers of the framework. In the Core Layer, designers provide reusable satisficing logic; in the Semi-Core Layer, decision-makers translate structural feedback behavior into decision formulations; and in the Application Layer, designers evaluate rural test problems under different policy conditions. Designers describe each layer in Sections 3.4, 3.5, and 3.6, respectively.

3.4 Core Layer: Unified Satisficing Architecture

3.4.1 Purpose of the Core Layer

In the Core Layer, designers establish the universal reasoning structure of the Context-Aware Control System. This layer contains the reusable satisficing principles, decision rules, and feedback interpretation mechanisms that remain applicable across multiple rural domains.

The Core Layer directly addresses F1, the problem framing functionality, and F3, the satisficing decision functionality, because it defines the unified operational philosophy that supports context-aware decision-making under uncertainty and resource constraints. Rather than developing separate control systems for each application domain, designers use the Core Layer to provide a common operational foundation that preserves consistency across the energy, water, and healthcare test problems examined in Chapter 4.

3.4.2 Satisficing Engine

Designers use the satisficing engine as the central decision mechanism within the framework. Through this engine, designers evaluate competing operational priorities and select actions that satisfy acceptable performance thresholds under active constraints. The purpose of the satisficing engine is not to maximize a single objective. Instead, the engine balances:

- operational continuity,
- resource preservation,
- stakeholder requirements,
- and degradation effects within a constrained decision environment.

This distinction becomes particularly important within rural cyber-physical-social systems because optimization assumptions frequently become unrealistic under unstable operating conditions. A mathematically optimal solution may depend upon ideal infrastructure, perfect forecasting, or complete information that does not exist

within the actual operating environment. The satisficing engine therefore prioritizes decisions that:

- remain feasible,
- preserve system viability,
- and maintain acceptable operational performance under changing conditions.

Within the framework, the satisficing engine evaluates:

- decision variables,
- operational constraints,
- degradation penalties,
- weighted deviations,
- and structural trade-offs.

The resulting solution represents an acceptable operating condition rather than a globally optimal configuration.

3.4.3 Closed-Loop Operational Evaluation

The Context-Aware Control System operates through repeated closed-loop decision cycles. After each operational decision is executed, the resulting system state is externally updated and re-evaluated within the next decision cycle. Through this repeated process, designers examine how operational policies influence long-term structural system behavior over time. The purpose of the closed-loop process is not autonomous learning or data-driven adaptation. Instead, through the repeated execution cycle, designers evaluate whether satisficing policies remain structurally acceptable under evolving operating conditions.

Through repeated policy execution, designers evaluate:

- whether short-term corrective actions introduce delayed degradation,

- whether resource depletion accelerates across repeated cycles,
- whether balancing and reinforcing feedback structures remain stable,
- and whether operational trade-offs remain acceptable under changing constraints.

This repeated evaluation process preserves consistency between the identified systems archetype and the resulting operational behavior. The framework therefore treats the operational decision process as an evolving structural system rather than as a single static optimization problem. Designers apply this closed-loop evaluation procedure consistently across the three test problems in Chapter 4. Each test problem solves a daily compromise Decision Support Problem, updates the relevant state variables externally between days, and re-solves the cDSP on the next day using the updated state.

3.4.4 Reuse Heuristics

Within the Core Layer, designers maintain reusable operational heuristics that remain applicable across multiple rural domains. These heuristics preserve structural consistency and operational coherence across repeated decision cycles.

Examples include:

- prioritizing fallback actions under uncertainty,
- preserving locally executable interventions,
- requiring operational feasibility under weak infrastructure,
- and recording the structural consequences of repeated corrective actions.

Designers apply these principles consistently across domains in order to preserve the unified satisficing philosophy of the framework. In the test problems of Chapter 4, designers therefore use the same satisficing engine, the same closed-loop evaluation procedure, and the same compromise Decision Support Problem

structure across the energy, water, and healthcare domains. Only the domain-specific decision variables, constraints, and goal equations differ across test problems.

3.5 Semi-Core Layer: Structural Translation and Constraint Mapping

3.5.1 Purpose of the Semi-Core Layer

Designers define the Semi-Core Layer as the structural translation layer of the Context-Aware Control System. Within this layer, decision-makers translate qualitative systems reasoning into operational decision formulations that can later support satisficing policy evaluation under changing rural operating conditions. The purpose of this layer is not merely to describe system behavior qualitatively. Instead, designers use the Semi-Core Layer to establish a structured pathway through which systems thinking becomes operationally actionable within the framework. The Semi-Core Layer directly addresses F2, the translation functionality, because it establishes the translation mechanism between causal loop representations, systems archetypes, and compromise Decision Support Problem formulations. This layer therefore represents one of the primary theoretical contributions of the thesis because it bridges the gap between descriptive systems interpretation and operational decision-making.

Existing systems thinking approaches frequently stop at qualitative interpretation. Researchers often use causal loop diagrams and systems archetypes to explain recurring structural behavior, but they rarely translate those structures into formal decision constructs capable of guiding operational policy evaluation. Within this thesis, designers extend beyond descriptive interpretation by operationalizing the structural behavior identified within the archetypes. Designers organize the Semi-Core Layer around four major activities. First, designers construct causal loop diagrams in order to identify the dominant feedback relationships governing system

behavior. Second, designers interpret these feedback structures through systems archetypes in order to identify recurring structural tensions. Third, designers quantify the interpreted archetypal behavior by translating balancing and reinforcing loops into operational decision constructs. Finally, decision-makers formulate the resulting operational problem as a compromise Decision Support Problem in which structural tensions become weighted satisficing trade-offs. The Semi-Core Layer therefore acts as the conceptual bridge between systems interpretation and satisficing operational execution. Through the Core Layer, designers provide the universal satisficing philosophy of the framework, and through the Application Layer, designers evaluate operational policies across generated rural scenarios. It is through the Semi-Core Layer, however, that designers determine how structural behavior itself becomes represented within the decision process. For this reason, designers consider this layer central to the operationalization of the Context-Aware Control System.

3.5.2 Construction of Causal Loop Diagrams (Method Step 2)

After defining the rural problem space through the Dilemma Triangle Method, designers construct causal loop diagrams (CLDs). The author draws the causal relationships in each causal loop diagram from established systems archetype and system dynamics literature (Senge, 1990; Braun, 2002; Libey et al., 2022), where researchers document comparable feedback structures as causal mechanisms in resource and service systems. In this thesis, the author uses causal readings from prior literature as modeling assumptions rather than independently validating causality for the specific rural systems examined in Chapter 4. In Section 5.1.7, the author states this assumption explicitly. Decision-makers use a causal loop diagram to represent a system's structure as a set of variables connected by causal links, organized into balancing loops that stabilize behavior and reinforcing loops that amplify it. Designers construct these diagrams in order to identify the dominant feedback relationships governing operational behavior within the system under

study. Designers use causal loop diagrams because rural cyber-physical-social systems rarely behave through isolated linear cause-and-effect relationships. Instead, system behavior emerges through interacting feedback processes that evolve across repeated operational cycles. The causal loop representation allows designers to move beyond isolated operational variables and instead examine the structural behavior of the complete system over time. Within the framework, designers use the causal loop diagrams to identify reinforcing loops, balancing loops, delayed degradation effects, resource interactions, operational dependencies, and recurring corrective actions that influence long-term system behavior. Designers construct the causal loop diagrams by identifying the major operational variables associated with the generated rural test problem. These variables differ across domains but typically include resource availability, operational demand, infrastructure condition, stakeholder pressure, degradation effects, corrective interventions, and long-term system capacity. After identifying the variables, designers define the causal relationships connecting them and determine whether the relationships reinforce or balance system behavior across time. Designers select which loops to retain in the diagram by tracing them back to the tension matrix and dilemmas identified in Method Step 2: a candidate loop earns a place in the CLD only if it visibly drives the recurring operational dilemma under study, so that a loop with no measurable effect on the decision variables central to that dilemma is left out even if it is technically present in the system.

Reinforcing loops represent structural conditions in which operational changes amplify themselves across repeated cycles. Decision-makers observe that these loops frequently generate accelerating growth, increasing degradation, or escalating dependence upon corrective actions. Balancing loops represent stabilizing relationships that attempt to preserve operational equilibrium or constrain excessive system behavior. However, balancing loops may also introduce delayed effects that

remain invisible during early operational stages. The causal loop representation becomes particularly important within rural systems because many structural failures do not emerge immediately. For example, an operational policy may initially improve service continuity while simultaneously weakening long-term system capacity. Similarly, repeated resource extraction may appear operationally acceptable during early cycles while gradually accelerating future depletion. These delayed structural effects frequently remain hidden when decision-makers examine isolated operational states rather than complete feedback structures. For this reason, designers use causal loop diagrams not simply as descriptive illustrations but as structural diagnostic tools that reveal the underlying feedback behavior governing the system. Designers use these diagrams to support systems archetype interpretation and operational translation within the framework. Designers demonstrate this step explicitly in Sections 4.1.2, 4.2.2, and 4.3.2 of Chapter 4, where designers construct causal loop diagrams for the photovoltaic battery microgrid, the rural water allocation problem, and the rural healthcare triage problem, respectively.

3.5.3 Archetype Interpretation (Method Step 3)

After constructing the causal loop diagrams, designers interpret the resulting feedback structures through systems archetypes. Designers use systems archetypes because recurring operational failures within rural cyber-physical-social systems rarely emerge as completely unique phenomena. Instead, many operational problems exhibit recurring structural patterns that repeatedly appear across multiple domains despite differences in application context. Within the thesis, designers examine archetypes including Fixes That Fail, Tragedy of the Commons, Shifting the Burden, Limits to Growth, and combined archetype interactions. Designers do not interpret these archetypes as domain-specific operational failures. Instead, designers interpret them as generalized structural patterns that explain how

reinforcing and balancing feedback relationships produce recurring system behavior across time.

Decision-makers use the Fixes That Fail archetype to characterize situations in which short-term corrective actions reduce immediate operational pressure while simultaneously introducing delayed degradation effects that weaken future system performance. Within rural environments, such behavior frequently appears when emergency operational interventions improve short-term service continuity but accelerate long-term infrastructure deterioration or resource depletion. Designers apply this archetype to the photovoltaic battery microgrid in Section 4.1, where repeated battery discharge reduces immediate service shortfalls but accelerates battery degradation.

Decision-makers use the Tragedy of the Commons archetype to characterize distributed overuse of a shared resource environment. Within this structure, multiple stakeholders independently pursue locally acceptable outcomes while collectively weakening the long-term sustainability of the shared system. Designers observe that this archetype frequently emerges within rural water systems, shared energy infrastructure, and collectively used environmental resources. Designers apply this archetype to the rural water allocation problem in Section 4.2, where each stakeholder group reduces its own service shortfall through additional withdrawal while collective withdrawal accelerates aquifer depletion.

Decision-makers use the Shifting the Burden archetype to characterize dependence upon symptomatic interventions that reduce immediate operational pressure while weakening the long-term capability of the fundamental solution. This behavior frequently appears when systems repeatedly prioritize emergency operational relief while neglecting structural preservation or preventive capacity development. Shifting the Burden differs structurally from Fixes That Fail in that it requires two distinct solution paths rather than one: a symptomatic relief loop and a separate

fundamental-solution loop that atrophies from disuse as the symptomatic path is favored. Fixes That Fail, by contrast, involves a single corrective action whose own side effect erodes the same capacity that the action depends on, with no competing fundamental solution present. Designers apply this archetype to the rural healthcare triage problem in Section 4.3, where repeated allocation of clinic capacity to symptomatic treatment reduces immediate acute backlog but weakens long-term community health capability.

Decision-makers use the Limits to Growth archetype to characterize reinforcing operational growth processes that eventually encounter hidden balancing

constraints. Early operational improvements initially appear successful, but delayed structural limitations gradually restrict further improvement and eventually destabilize the system. Designers include this archetype in the framework for completeness and reserve its operational application to future combined-archetype test problems beyond the present thesis.

Designers use these archetypes to interpret the dominant structural tensions governing operational behavior within each generated test problem. The purpose of the archetype interpretation stage is not simple classification. Instead, designers use the archetypes to determine how structural behavior should later influence operational decision-making within the satisficing framework.

3.5.4 Archetype Quantification (Method Step 4 - Translation)

After identifying the governing archetype, decision-makers translate the structural behavior into operational decision constructs through a process that designers define as archetype quantification. This stage represents one of the primary conceptual contributions of the thesis because it transforms qualitative systems interpretation into operational decision structures capable of supporting satisficing policy evaluation. Designers do not treat the archetypes as purely descriptive

narratives. Instead, designers use them to define how structural feedback behavior should influence operational decision-making. Within this translation process, balancing relief loops become operational service objectives, reinforcing erosion loops become degradation penalties, structural tensions become weighted trade-offs, and delayed effects become repeated closed-loop operational constraints.

Within the Fixes That Fail archetype examined in Section 4.1, the balancing relief loop represents the operational objective of reducing immediate service shortfalls, while the reinforcing erosion loop represents delayed degradation effects that weaken long-term system viability. In the photovoltaic battery microgrid, the balancing loop describes how increased battery discharge reduces stakeholder service shortfalls, while the reinforcing loop describes how repeated discharge accelerates battery degradation and reduces future usable capacity. Designers therefore translate the balancing loop into a service-target goal equation for each stakeholder group and the reinforcing loop into a battery feasibility constraint that depends on accumulated degradation across cycles.

Within the Tragedy of the Commons archetype examined in Section 4.2, multiple balancing loops act simultaneously, with one balancing loop per stakeholder group. Each balancing loop describes how an increase in that group's withdrawal reduces its own service shortfall. A single reinforcing loop describes how total community withdrawal lowers the aquifer level and reduces usable groundwater capacity for every group. Designers therefore translate the balancing loops into one service-target goal equation per stakeholder group and the reinforcing loop into a sum constraint on total community withdrawal together with an aquifer state update across cycles.

Within the Shifting the Burden archetype examined in Section 4.3, two balancing loops act simultaneously, but they operate on different time scales. The symptomatic relief loop describes how increased capacity allocated to symptomatic

treatment reduces immediate acute backlog. The fundamental solution loop describes how increased capacity allocated to preventive care strengthens long-term community health capability and reduces the future incidence of preventable conditions. A reinforcing erosion loop describes how repeated reliance on symptomatic treatment displaces preventive activity and eventually increases future acute caseload. Designers therefore translate the two balancing loops into two goal equations (a symptomatic-response goal and a preventive-coverage goal) and the reinforcing erosion loop into a state update for community health capability that drives the day-to-day acute presentation estimate.

Through this process, designers operationalize systems thinking within the satisficing control framework. Designers use archetype quantification to preserve structural consistency between the identified feedback behavior and the resulting operational decisions.

3.5.5 Constraint Mapping and cDSP Translation (Method Step 5)

After interpreting and quantifying the archetypal structure, decision-makers formulate the operational problem as a compromise Decision Support Problem. Within this formulation, designers define the Given conditions, Find variables, Satisfy constraints, and Minimize objectives associated with the operational environment. The purpose of the cDSP formulation is not to generate globally optimal solutions under ideal assumptions. Instead, designers use the cDSP structure to represent operational trade-offs among competing structural priorities under uncertainty and resource constraints. Through this formulation, decision-makers translate structural tensions into operational decision variables, degradation penalties, weighted deviations, operational thresholds, and satisficing trade-offs. The Given conditions define the operating assumptions associated with the generated rural scenario, including resource availability, stakeholder requirements, environmental conditions, degradation states, and infrastructure limitations. The

Find variables represent the operational actions or allocation decisions that must be determined within the satisficing process. The Satisfy conditions define the structural constraints that preserve operational feasibility and long-term viability. Finally, decision-makers use the Minimize formulation to represent the weighted deviations associated with competing operational priorities and structural tensions. Within this process, designers preserve consistency between the identified archetype, the causal structure, and the resulting operational formulation. Designers do not allow the operational model to behave independently from the structural interpretation of the system: concretely, every decision variable, constraint, and deviation term that appears in the cDSP must trace back to a specific loop or structural tension identified during the CLD construction and archetype interpretation stages, so that no goal, constraint, or penalty is introduced into the Satisfy or Minimize blocks without a corresponding structural justification from Method Steps 3 and 4. Instead, designers ensure that the operational decision process remains structurally aligned with the governing feedback behavior identified through the causal loop analysis and archetype interpretation stages.

In each of the three test problems in Chapter 4, designers present the cDSP formulation in two complementary forms. First, designers provide a CLD-to-cDSP mapping table that shows how each variable identified in the causal loop diagram appears as a decision variable, state variable, constraint, deviation term, or goal equation in the quantitative model. Second, designers present the formulation in its mathematical form using a Given, Find, Satisfy, Minimize block. This presentation style preserves consistency across the three test problems and allows designers to compare archetypal feedback structures within a common decision-formulation language.

3.6 Application Layer: Execution and Policy Evaluation

Decision-makers use Frame to identify the structural tension in a rural cyber-physical-social situation, and use Decide to formulate and solve the operational decision within that frame. Designers define the Application Layer as the operational execution and evaluation layer of the Context-Aware Control System. Within this layer, designers apply the structural reasoning developed in the Core and Semi-Core Layers to specific rural operating environments in order to evaluate satisficing policies under evolving conditions. While designers use the Core Layer to establish the universal satisficing philosophy of the framework, and use the Semi-Core Layer to translate structural feedback behavior into operational decision formulations, designers use the Application Layer to evaluate how these formulations behave under realistic rural conditions across repeated operational cycles. Designers do not treat this layer as a simple implementation stage in which operational decisions are mechanically executed. Instead, designers treat it as the operational environment in which structural policies interact with changing conditions, stakeholder priorities, resource limitations, and degradation behavior across time.

Within rural cyber-physical-social systems, operational conditions rarely remain stable. Resource availability changes unpredictably, infrastructure conditions degrade unevenly, environmental disturbances emerge unexpectedly, and stakeholder priorities evolve across repeated decision cycles. Under such conditions, a single operational decision cannot fully characterize the long-term behavior of the system. Designers therefore organize the Application Layer around repeated operational evaluation rather than one-time policy execution. Within the Application Layer, designers organize the operational workflow into five sequential stages: Sense, Frame, Decide, Act, and Learn. Designers do not treat these stages as isolated computational modules. Instead, they represent repeated operational reasoning processes through which designers preserve structural consistency

between the interpreted systems archetype, the satisficing decision formulation, and the resulting operational behavior. Designers therefore use the Application Layer as the environment in which designers evaluate whether structurally informed satisficing policies remain operationally feasible and acceptable under changing rural conditions.

3.6.1 Sense: Establishing the Operating Context

Within the Application Layer, designers begin by defining the operating conditions associated with each generated rural scenario. The Sense stage establishes the active operational environment within which satisficing decisions must later be evaluated. Designers do not treat sensing as a purely technical data collection activity because rural cyber-physical-social systems frequently operate under incomplete information, intermittent infrastructure, and inconsistent communication capability. Instead, designers define sensing as the broader process through which designers establish the current operational state of the rural environment. This is the specific point at which the framework becomes context-aware in the operational sense: the Sense stage is what supplies the context (the current resource state, stakeholder priorities, and environmental conditions) that the Core Layer's satisficing engine and the Semi-Core Layer's translation procedure use to produce a decision suited to this particular operating moment rather than a fixed, context-independent rule.

The Sense stage therefore includes both technical and contextual operating conditions. These conditions include resource availability, stakeholder priorities, environmental assumptions, infrastructure limitations, operational demand levels, degradation states, and uncertainty conditions associated with the scenario under study. In many rural environments, operational information may originate from incomplete sensor data, locally reported observations, practical field experience, or partially available infrastructure signals. For this reason, designers do not assume

that all operational information remains fully observable or continuously available. Instead, designers structure the framework so that operational reasoning can continue even when portions of the information environment remain uncertain or partially unavailable. The purpose of the Sense stage is therefore not to construct a perfect representation of the operating environment. Rather, designers use this stage to establish the feasible operational conditions within which satisficing decisions must later be evaluated.

Designers apply the Sense stage to each of the three rural test problems in Section 4.1 through Section 4.3, where designers describe the specific operating conditions sensed within the photovoltaic battery microgrid, rural water allocation, and healthcare triage problems, respectively.

3.6.2 Frame

After establishing the operating conditions, designers frame the operational state by combining structural information from the Semi-Core Layer with the active operational constraints established in the Sense stage. The Frame stage represents the point at which decision-makers confirm which governing systems archetype applies to the currently sensed conditions, and identify how the structural tension that archetype describes, the relevant balancing and reinforcing loops, is manifesting in this specific operating period. Designers do not treat this as a numerical calculation; it is a structural recognition step that determines which frame the decision that follows must be formulated within. Designers use the structural information the Semi-Core Layer provides to confirm consistency between the currently sensed conditions and the governing systems archetype. Designers apply the Frame stage to each of the three test problems in Section 4.1 through Section 4.3, where designers confirm the governing archetype, Fixes That Fail, Tragedy of the Commons, or Shifting the Burden, that applies to the photovoltaic battery microgrid, rural water allocation, and healthcare triage problems, respectively.

3.6.3 Decide

After framing the situation, decision-makers formulate and solve the daily compromise Decision Support Problem to determine the specific satisficing action. The Decide stage represents the point at which the structural recognition from the Frame stage becomes a quantitative decision: decision-makers evaluate competing operational actions through structurally informed satisficing trade-offs rather than isolated optimization objectives. During this stage, decision-makers examine how alternative actions influence operational continuity, degradation behavior, stakeholder satisfaction, and long-term system viability across repeated operational cycles, and evaluate weighted satisficing trade-offs associated with stakeholder priorities, operational feasibility, degradation penalties, and infrastructure limitations. Decision-makers use the cDSP formulation developed in the Semi-Core Layer to determine which operational actions remain structurally acceptable under the active scenario conditions. Decision-makers apply the Decide stage to each of the three test problems in Section 4.1 through Section 4.3, where decision-makers solve the daily cDSP for the photovoltaic battery microgrid, rural water allocation, and healthcare triage problems, respectively

3.6.4 Act

After evaluating the operational trade-offs, decision-makers execute a satisficing action that preserves operational feasibility while remaining structurally consistent with the governing archetype. The purpose of the Act stage is not simply to implement the numerically best decision. Instead, decision-makers execute policies that remain operationally feasible within the active rural environment while preserving acceptable long-term system behavior. Within rural cyber-physical-social systems, operational feasibility frequently becomes more important than

strict numerical optimality. A theoretically superior decision may become impossible to implement because of weak infrastructure, limited manpower, degraded communication capability, or insufficient operational resources. For this reason, designers ensure that the executed operational policy remains compatible with local operating conditions and resource constraints.

Designers apply the Act stage to each of the three test problems in Section 4.1 through Section 4.3, where designers describe the specific executed policy within the photovoltaic battery microgrid, rural water allocation, and healthcare triage problems, respectively.

3.6.5 Learn

After policy execution, designers evaluate how the operational behavior evolves across repeated decision cycles under changing conditions. Designers define the Learn stage as the repeated structural evaluation process through which decision-makers compare system behavior across multiple operating scenarios, weight configurations, stakeholder priorities, and resource conditions.

Designers do not treat the operational decision process as a static optimization problem solved only once. Instead, designers evaluate how operational policies influence long-term structural behavior across repeated execution cycles. This stage therefore preserves consistency between the interpreted systems archetype and the evolving operational behavior of the system. Within the Learn stage, designers evaluate degradation behavior, operational continuity, trade-off stability, acceptable operating regions, and long-term resource preservation under changing conditions. Decision-makers compare how different satisficing policies influence system performance across repeated operational cycles and determine whether the selected policies remain structurally acceptable over time.

Designers apply the Learn stage to each of the three test problems in Section 4.1 through Section 4.3, where designers evaluate how the executed policy performs across the full 30-day decision horizon within the photovoltaic battery microgrid, rural water allocation, and healthcare triage problems, respectively.

3.7 Operational Workflow of the Framework

The complete operational workflow of the Context-Aware Control System consolidates the architectural layers described in Sections 3.4 through 3.6 into a sequence of eight method steps. The workflow begins with qualitative problem framing and ends with repeated satisficing policy evaluation. Designers apply the same workflow, unmodified, to every test problem in Chapter 4. The steps are as follows.

Step 1. Frame the rural problem space using the Dilemma Triangle Method. Decision-makers identify the dominant tensions among local knowledge, uncertainty, and limited resources for the specific rural domain under study. This step establishes the structural conditions that motivate the operational decision process and corresponds to Section 3.2 of this chapter.

Step 2. Construct a causal loop diagram for the rural problem under study. Decision-makers identify the operational variables, infrastructure conditions, stakeholder pressures, and feedback relationships that govern long-term system behavior, and label balancing and reinforcing loops within the diagram. This step corresponds to Section 3.5.2 of this chapter and to Sections 4.1.2, 4.2.2, and 4.3.2 of Chapter 4.

Step 3. Interpret the causal loop diagram through a systems archetype. Decision-makers identify the recurring structural pattern, Fixes That Fail, Tragedy of the Commons, Shifting the Burden, or Limits to Growth, that governs the interaction

between the balancing and reinforcing loops. This step corresponds to Section 3.5.3 of this chapter and to Sections 4.1.3, 4.2.3, and 4.3.3 of Chapter 4.

Step 4. Quantify the archetypal feedback structure. Decision-makers translate balancing relief loops into operational service objectives, reinforcing erosion loops into degradation penalties or feasibility constraints, structural tensions into weighted trade-offs, and delayed effects into repeated closed-loop operational constraints. This step corresponds to Section 3.5.4 of this chapter and is applied within the cDSP formulations of Sections 4.1.4, 4.2.4, and 4.3.4 of Chapter 4.

Step 5. Formulate the operational problem as a compromise Decision Support Problem. Decision-makers define the Given, Find, Satisfy, and Minimize blocks that describe the decision environment, documenting the translation explicitly through a CLD-to-cDSP mapping table for each test problem. This step corresponds to Section 3.5.5 of this chapter and is applied within Sections 4.1.4, 4.2.4, and 4.3.4 of Chapter 4.

Step 6. Execute the satisficing policy on day k . Decision-makers solve the daily cDSP using the active state variables and operating conditions established through the Sense stage. The satisficing engine's output specifies the daily allocation, withdrawal, or staff-hour assignment that preserves operational feasibility while respecting the structural tension identified through the archetype. This step corresponds to Sections 3.4.2, 3.6.2, and 3.6.3 of this chapter.

Step 7. Update the state variables externally and re-evaluate the system. Decision-makers update the state variables that carry structural memory across cycles: battery state of charge and state of health in the Fixes That Fail formulation of Section 4.1, aquifer state of charge and usable capacity in the Tragedy of the Commons formulation of Section 4.2, and community health capability in the Shifting the Burden formulation of Section 4.3. Decision-makers then return to Step 6 with the

updated state and re-solve the cDSP for day $k+1$, repeating the cycle across the decision horizon. This step corresponds to Sections 3.4.3 and 3.6.4 of this chapter.

Step 8. Compare baseline and CACCS policies across the decision horizon. Designers compare policy behavior across all three test problems in Chapter 4 to show how decision-makers use the method structure consistently across distinct rural domains. Through this eight-step workflow, decision-makers and designers move from qualitative interpretation of a rural decision environment to an operationally executable satisficing policy that respects the structural behavior of the governing systems archetype. In Chapter 4, the author demonstrates the workflow in detail across three test problems.

Decision-makers satisfy F1, the problem framing functionality, by using Steps 1 through 3 to frame the rural problem space, construct the causal loop diagram, and interpret the governing systems archetype. Decision-makers satisfy F2, the translation functionality, by using Steps 4 and 5 to translate the frame into cDSP decision constructs. Decision-makers satisfy F3, the satisficing decision functionality, by using Step 6 to solve the daily satisficing decision problem. Decision-makers satisfy F4, the closed-loop execution functionality, by using Step 7 to update state variables and reformulate the cDSP. Designers satisfy F5, the cross-domain demonstration functionality, by using Step 8 and the three test problems of Chapter 4 to compare method use across energy, water, and healthcare domains.

CHAPTER 4

Test Problems for the Context-Aware Control System Framework

In this chapter, designers apply the Context-Aware Control System method developed in Chapter 3 to three rural test problems. Designers use these test problems to demonstrate that decision-makers can translate structural feedback patterns into operational decision formulations across distinct systems archetypes and distinct rural service domains. Designers follow the eight-step workflow described in Section 3.7 throughout this chapter. First, designers represent each rural problem through a causal loop diagram (Method Step 2, Section 3.5.2). Second, designers interpret the causal structure through a systems archetype (Method Step 3, Section 3.5.3). Third, designers quantify the archetypal feedback structure into operational decision constructs (Method Step 4, Section 3.5.4). Fourth, decision-makers translate the result into a compromise Decision Support Problem formulation (Method Step 5, Section 3.5.5). Finally, designers describe the closed-loop policy evaluation procedure (Method Steps 6 through 8, Sections 3.6 and 3.4.3) used to compare baseline and CACCS satisficing policies across repeated operational cycles.

Section 4.1 examines a photovoltaic battery microgrid governed by the Fixes That Fail archetype. Section 4.2 examines a rural water allocation problem governed by the Tragedy of the Commons archetype. Section 4.3 examines a rural healthcare triage problem governed by the Shifting the Burden archetype. Together, these test problems demonstrate that the framework generalizes across three distinct archetypal feedback structures and across three rural service domains (energy, water, and healthcare).

Designers deliberately test the framework on three single-archetype problems rather than on one combined problem because the two designs answer different validation questions. A single comprehensive test problem in which multiple archetypes interact simultaneously would demonstrate that the framework can handle archetype combination, but it would not by itself demonstrate that the translation method in Section 3.5.4 generalizes across structurally distinct feedback patterns, since any single problem instantiates only one combined structure. Holding the domain and the archetype separately distinct across three test problems isolates the archetype as the controlled variable: the same eight-step method, the same Core Layer satisficing engine, and the same Semi-Core Layer translation procedure are applied to three governing structures that differ in the number and arrangement of balancing and reinforcing loops (a single reinforcing erosion loop in Fixes That Fail, three parallel balancing loops sharing one reinforcing loop in Tragedy of the Commons, and two balancing loops operating on different time scales in Shifting the Burden). This design directly supports the Empirical Structural Validity claim of Section 4.4.2, which rests on the three archetypes being structurally distinct rather than superficially relabeled. Combining archetypes within a single problem is identified as a specific direction for future work in Section 5.3, where a fourth test problem under the Limits to Growth archetype and combined-archetype formulations are proposed as the next extension once single-archetype translation has been established.

Applying the Sense, Frame, Decide, Act, and Learn Workflow Across the Three Test Problems

In Chapter 4, designers examine each rural service situation by first framing it as Fixes That Fail, Tragedy of the Commons, or Shifting the Burden, rather than treating it as an ordinary allocation problem.

Sections 3.6.1 through 3.6.5 describe the Sense, Frame, Decide, Act, and Learn stages of the Application Layer in general, domain-independent terms. Here, designers apply each stage explicitly to the three test problems developed in this chapter, so that the general method description in Chapter 3 and its domain-specific application remain clearly separated while still being easy to connect to one another.

Sense

Within the test problems of Chapter 4, the sensed operating conditions differ across domains. In the photovoltaic battery microgrid problem (Section 4.1), the Sense stage includes available stored energy, fluctuating solar generation, demand levels across stakeholder groups, and current degradation conditions associated with the battery system. Within the rural water allocation problem (Section 4.2), the sensed conditions include groundwater levels, withdrawal demand, recharge uncertainty, and competing user priorities. Within the healthcare triage problem (Section 4.3), the operating state includes total daily clinic capacity, current acute presentations, the minimum administrative continuity allocation, and the current community health capability state carried forward from the previous decision cycle. Designers therefore use the Sense stage to establish the active operational context required for the later stages of satisficing decision evaluation.

Frame

In Section 4.1, the interpreted structure corresponds to Fixes That Fail, so designers evaluate whether immediate corrective actions (battery discharge) introduce delayed degradation effects that weaken future operational capability. In Section 4.2, the interpreted structure corresponds to Tragedy of the Commons, so designers evaluate how distributed allocation decisions across the three stakeholder groups collectively influence long-term aquifer preservation. In Section 4.3, the interpreted

structure corresponds to Shifting the Burden, so designers examine whether symptomatic operational relief weakens the long-term capability of the fundamental solution (preventive care). Within Limits to Growth structures, designers would evaluate whether reinforcing operational improvements eventually encounter hidden balancing constraints that restrict future system performance.

Decide

Within each test problem, decision-makers formulate and solve the daily compromise Decision Support Problem to determine the specific satisficing action consistent with the interpreted archetype. Within the photovoltaic battery microgrid problem (Section 4.1), decision-makers solve the daily cDSP to determine the specific energy allocation across the three stakeholder groups that preserves battery feasibility while respecting the Fixes That Fail structure. Within the rural water allocation problem (Section 4.2), decision-makers solve the daily cDSP to determine the specific withdrawal volumes for the domestic, agricultural, and enterprise groups that remain within the aquifer feasibility constraint while respecting the Tragedy of the Commons structure. Within the healthcare triage problem (Section 4.3), decision-makers solve the daily cDSP to determine the specific staff-hour allocation among symptomatic treatment, preventive care, and administrative continuity that respects the Shifting the Burden structure. The Decide stage therefore represents the point at which the archetype recognized during the Frame stage becomes a specific, quantitative decision that the Act stage then executes.

Act

The executed action depends upon the generated rural scenario and the governing archetype. Within the photovoltaic battery microgrid problem (Section 4.1), the executed action involves daily energy allocation decisions across three stakeholder

groups under fluctuating generation conditions, with battery discharge bounded by the current state of charge and usable capacity. Within the rural water allocation problem (Section 4.2), the executed policy assigns daily withdrawal volumes to the domestic, agricultural, and enterprise groups subject to the aquifer feasibility constraint. Within the healthcare triage problem (Section 4.3), the executed action allocates daily staff-hours among symptomatic treatment, preventive care, and administrative continuity activities subject to the total clinic capacity and minimum continuity allocation. Designers ensure that the executed policies remain structurally aligned with the interpreted archetype. Within the Fixes That Fail structure of Section 4.1, designers evaluate whether the selected corrective action accelerates future degradation. Within the Tragedy of the Commons structure of Section 4.2, designers evaluate whether the daily collective withdrawal weakens shared resource sustainability. Within the Shifting the Burden structure of Section 4.3, designers evaluate whether the daily allocation to symptomatic treatment displaces enough preventive activity to weaken long-term community health capability. The Act stage therefore represents the operational realization of the structurally informed satisficing policy selected during the Decide stage.

Learn

Within the photovoltaic battery microgrid system (Section 4.1), designers evaluate how repeated deep discharge behavior influences long-term battery degradation and future operational capability. Within the rural water allocation problem (Section 4.2), designers evaluate how repeated collective withdrawal policies influence groundwater depletion and long-term sustainability. Within the healthcare triage problem (Section 4.3), designers examine whether repeated prioritization of symptomatic treatment weakens preventive capability and increases long-term system dependence upon symptomatic intervention. The Learn stage therefore allows designers to evaluate not only whether a policy performs

adequately during a single operational cycle, but also whether it preserves structurally acceptable system behavior across time. Designers also use this stage to compare baseline operational policies against archetype-aware satisficing policies in order to evaluate whether structurally informed decision-making improves long-term operational stability under uncertainty and resource constraints.

4.1 Photovoltaic Battery Microgrid under the Fixes That Fail Archetype

Designers examine a photovoltaic battery microgrid that supplies electricity to three stakeholder groups in a rural community. The first group consists of critical services such as clinic refrigeration and essential lighting. The second group consists of household electricity use. The third group consists of productive activities such as small shops and milling operations. The microgrid operator allocates available energy each day across these three groups subject to a battery state of charge and a battery state of health that updates across the decision horizon. When the operator increases battery discharge, delivered energy increases and service shortfalls decrease in the short term; however, repeated discharge accelerates battery degradation, reduces battery state of health, reduces usable capacity, and increases the difficulty of meeting demand in later periods. Designers interpret this interaction through the Fixes That Fail archetype.

Designers have presented the full development of this test problem in a separate paper, where designers describe the problem context (Section 3.1 of that paper), construct the causal loop diagram representing the Fixes That Fail structure (Section 3.2, Method Step 1), interpret the structure through the archetype (Section 3.3, Method Step 2), and translate it into a compromise Decision Support Problem (Section 3.4, Method Step 3). Designers summarize the key elements of that

translation below for reference and follow the same template in Sections 4.2 and 4.3.

4.1.1 Summary of the Fixes That Fail Translation

In the causal loop diagram of the microgrid, designers identify one balancing loop B1 in which battery discharge increases delivered energy and reduces service shortfall, and one reinforcing loop R1 in which battery discharge accelerates battery degradation, reduces battery state of health, reduces usable capacity, increases future service shortfall, and increases operator pressure to discharge again. The interaction between B1 and R1 produces the Fixes That Fail structure. Within the cDSP formulation, designers represent the balancing loop through two-sided goal equations for each of the three stakeholder groups (critical services, households, and productive users) and the reinforcing loop through battery feasibility constraints tied to the state of charge and the state of health. Decision-makers solve a daily cDSP for each day $k = 1, 2, \dots, 30$, across seven priority-weight scenarios, and compare a baseline policy that constrains battery discharge only by physical feasibility against a CACCS policy that adds a fixed daily discharge ceiling in order to preserve future battery capability. Detailed results, including ternary plots of stakeholder allocation and battery state of health outcomes under all seven weight scenarios, are presented in Section 4.1.7.

4.1.2 Why This Test Problem Is Appropriate for Validation

The decision-maker first sees declining energy service. The decision-maker might frame the situation as insufficient energy allocation. The decision-maker should instead frame the situation as Fixes That Fail, because the decision-maker gains short-term relief by discharging the battery aggressively, while future decision-makers have less battery capability after repeated discharge. The photovoltaic battery microgrid is an appropriate vehicle for testing the Fixes That Fail archetype for three reasons. First, it is structurally canonical: Fixes That Fail is the archetype

designers use to classify situations in which a symptomatic intervention provides short-term relief while simultaneously weakening the structural condition that enables long-term performance. In the microgrid context, increasing daily energy delivery reduces immediate service shortfall but accelerates battery degradation through deep-discharge cycles. This is exactly the feedback structure designers expect from the Fixes That Fail archetype. Second, the microgrid domain is technically tractable. Battery state of charge dynamics are well-characterized, and the relationship between daily discharge depth and long-term capacity fade can be modeled with a physically motivated equation. Because of this tractability, designers can verify the archetype-predicted behavior against computational results rather than relying on qualitative interpretation alone. Third, designers find independent motivation for the microgrid domain in the rural CPSS literature. By reading Bertus et al. (2012), one understands that cDSP formulations can manage rural energy allocation under multiple competing loads. Designers extend that work through the present test problem by adding the archetype structure, the closed-loop state update, and the comparative policy evaluation that Bertus et al. did not provide.

4.1.3 Problem Context

A daily energy allocation problem is examined in a rural community served by an off-grid photovoltaic battery microgrid. The microgrid consists of a solar photovoltaic generation source and a battery storage system with a nominal capacity of 200 kWh. The operator distributes available energy across three priority categories: critical services such as clinic refrigeration and essential lighting, with a fixed daily demand of 12 kWh; household electricity use, with a fixed daily demand of 80 kWh; and productive enterprise activities such as small shops and milling operations, with a fixed daily demand of 50 kWh. Designers treat daily photovoltaic generation as a known, deterministic input for each day of the decision horizon rather than as a stochastic quantity; designers identify the stochastic

representation of solar variability as a limitation of the current formulation in Section 5.2. Total demand across the three categories persistently exceeds photovoltaic generation, so the operator draws on stored battery capacity each day to supplement the available energy supply.

When the microgrid operator increases battery discharge to meet immediate demand, delivered energy increases and service shortfall decreases through the symptomatic relief loop B1. However, repeated battery discharge increases battery degradation, which reduces the battery state of health. A lower state of health reduces the usable battery capacity available in future periods, since designers bound both the minimum reserve and the maximum usable charge as fractions of state of health (Equations 7 and 8 in Section 4.1.6). As usable capacity declines, the operator faces greater pressure to discharge the battery again in subsequent periods, which further accelerates degradation. Through this structural sequence, the system drifts toward reduced battery capability even when the operator is successfully meeting daily service targets in the short term. This is the structural tension that the Fixes That Fail archetype characterizes.

4.1.4 CLD Construction (Method Step 2)

Following Method Step 2 of Section 3.5.2, designers construct a causal loop diagram representing the feedback relationships among battery discharge, delivered energy to each service category, service shortfall per category, battery degradation, battery state of health, and usable battery capacity. The author presents the resulting diagram in Figure 4.1.

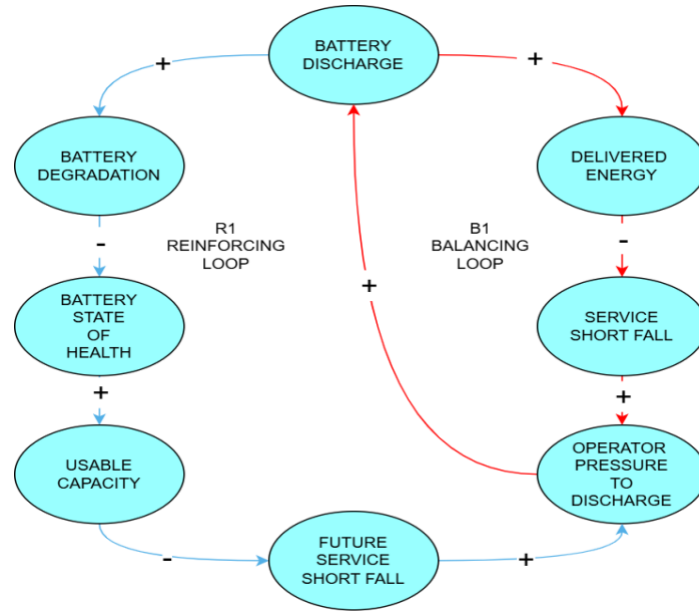


Figure 4.1. Causal loop diagram for the photovoltaic battery microgrid under the Fixes That Fail archetype. B1: balancing relief loop. R1: reinforcing erosion loop.

Designers identify two structural loops. The first is the balancing relief loop B1: when battery discharge increases, delivered energy increases and service shortfall decreases. This is a balancing loop that provides short-term relief. The second is the reinforcing erosion loop R1: when battery discharge increases, battery degradation increases, which reduces battery state of health, which reduces usable battery capacity, which increases future service shortfall and increases operator pressure to discharge the battery again. This is a reinforcing loop that amplifies long-term degradation.

4.1.5 Archetype Interpretation (Method Step 3)

Following Method Step 3 of Section 3.5.3, designers interpret the causal structure through the Fixes That Fail archetype. The balancing relief loop B1 corresponds to the symptomatic fix: increasing battery discharge reduces immediate shortfall. The reinforcing erosion loop R1 corresponds to the unintended consequence: increased discharge accelerates battery degradation, which reduces future capacity. The

structural tension between these two loops defines the Fixes That Fail pattern. Per the Fixes That Fail archetype, designers expect that an operator who responds only to immediate service shortfall, without accounting for the delayed reinforcing effect on battery state of health, will progressively erode battery capability. The short-term fix of aggressive discharge undermines the long-term structural condition that enables continued service delivery.

4.1.6 Archetype Quantification and cDSP Formulation (Method Steps 4 and 5)

In Table 4.1, the author presents the CLD-to-cDSP mapping for the photovoltaic battery microgrid.

Table 4.1. CLD-to-cDSP mapping for the photovoltaic battery microgrid.

CLD Variable	Role in Archetype	cDSP Element	Representation
Battery discharge	Corrective action (B1)	Find	Decision variable $E_k^{\{bat,dis\}}$
Delivered energy (critical, household, productive)	Balancing loop B1 outcome	Find	Allocation decision variables E_k^C, E_k^H, E_k^P
Service shortfall per category	Immediate service pressure	Goal equation	Two-sided deviation variables $d_{-,i,k}, d_{+,i,k}$ for i in $\{C,H,P\}$
Battery degradation	Reinforcing erosion loop (R1)	State update	Reduces SOH_k across cycles (external update)

Battery state of health SOH_k	State variable	Given / state update	Scales usable capacity via $SOC_{k, \min}$ and $SOC_{k, \max}$
Usable battery capacity	Long-term constraint on withdrawal	Battery feasibility constraint	$SOC_{k, \min} = 0.20 \times SOH_k \times E_{nom}$; $SOC_{k, \max} = 0.95 \times SOH_k \times E_{nom}$
Future service shortfall / operator pressure to discharge	Reinforcing loop R1 across cycles	Represented through objective weights and repeated solve	Weighted deviation objective Z_k , re-solved each day $k+1$
Discharge ceiling (CACCS only)	Structural constraint slowing R1	Satisfy	$E_k^{\{bat, dis\}} \leq B$ ($B = 50$ kWh/day)

Following Method Steps 4 and 5 of Sections 3.5.4 and 3.5.5, decision-makers translate the archetypal structure into operational decision constructs. The balancing relief loop B1 becomes a two-sided goal equation for each stakeholder group, with deviation variables that penalize both shortfall and excess delivery. The reinforcing erosion loop R1 becomes a pair of battery feasibility constraints in which the minimum reserve and the maximum usable charge both scale with the current battery state of health. The structural tension between short-term relief and long-term preservation becomes a weighted deviation objective evaluated across seven priority-weight combinations, and the delayed constraint becomes, in the

CACCS formulation only, a fixed daily discharge ceiling that slows the reinforcing erosion loop.

Decision-makers formulate the daily cDSP as follows. For each day $k = 1, 2, \dots, 30$:

Given: photovoltaic energy available on day k (E_k^{pv}), demand for critical services, households, and productive users on day k (D_k^C, D_k^H, D_k^P , fixed at 12, 80, and 50 kWh per day respectively), battery state of charge at the start of day k (SOC_k), battery state of health at the start of day k (SOH_k), nominal battery capacity ($E_{nom} = 200 \text{ kWh}$), targets equal to the corresponding daily demands ($T_k^C = D_k^C, T_k^H = D_k^H, T_k^P = D_k^P$), and priority weights w_C, w_H, w_P for the three service goals.

Assumptions: decision-makers solve one static compromise Decision Support Problem for each day. Designers treat photovoltaic availability and stakeholder demands as known for each day. Decision-makers update battery state of charge and state of health externally after each daily solve and use the updated values as inputs to the next day's decision problem. Designers use battery discharge as the corrective action that reduces immediate service shortfalls, and designers represent long-term erosion through battery feasibility constraints and state updates rather than as a separate goal.

Find: daily allocations to critical services, households, and productive users (E_k^C, E_k^H, E_k^P), the battery discharge and charging decisions ($E_k^{\{bat,dis\}}, E_k^{\{bat,ch\}}$), and the two-sided deviation variables for each stakeholder group ($d_{-C,k}, d_{+C,k}, d_{-H,k}, d_{+H,k}, d_{-P,k}, d_{+P,k}$).

Satisfy:

$$E_k^C + E_k^H + E_k^P + E_k^{bat,ch} \leq E_k^{pv} + E_k^{bat,dis} \quad (1)$$

$$0 \leq E_k^C \leq D_k^C \quad (2)$$

$$0 \leq E_k^H \leq D_k^H \quad (3)$$

$$0 \leq E_k^P \leq D_k^P \quad (4)$$

$$E_k^{bat,dis} \leq SOC_k - SOC_{k,min} \quad (5)$$

$$E_k^{bat,ch} \leq SOC_{k,max} - SOC_k \quad (6)$$

$$SOC_{k,min} = 0.20 \times SOH_k \times E_{nom} \quad (7)$$

$$SOC_{k,max} = 0.95 \times SOH_k \times E_{nom} \quad (8)$$

$$E_k^C + d_{C,k}^- - d_{C,k}^+ = D_k^C \quad (9)$$

$$E_k^H + d_{H,k}^- - d_{H,k}^+ = D_k^H \quad (10)$$

$$E_k^P + d_{P,k}^- - d_{P,k}^+ = D_k^P \quad (11)$$

$$d_{C,k}^-, d_{C,k}^+ \geq 0 \quad (12)$$

$$d_{H,k}^-, d_{H,k}^+ \geq 0 \quad (13)$$

$$d_{P,k}^-, d_{P,k}^+ \geq 0 \quad (14)$$

In the CACCS formulation, designers add a fixed daily discharge ceiling that directly targets the reinforcing erosion loop R1:

$$E_k^{bat,dis} \leq B \quad [CACCS \text{ only}] \quad (16)$$

where B is the acceptable daily battery discharge target. Designers set B = 50 kWh/day, following

Minimize:

$$Z_k = w_C^- d_{\{C,k\}}^- + w_C^+ d_{\{C,k\}}^+ + w_H^- d_{\{H,k\}}^- + w_H^+ d_{\{H,k\}}^+ + w_P^- d_{\{P,k\}}^- + w_P^+ d_{\{P,k\}}^+ \dots \dots \dots (15)$$

Following Method Step 7, decision-makers update battery state of charge and battery state of health externally after each daily solve, and designers re-solve the

cDSP on day $k+1$ using the updated state. The two scenarios evaluated are: Baseline (battery discharge constrained only by Equations 5 through 8) and CACCS (battery discharge additionally constrained by the discharge ceiling above). Designers evaluate both scenarios across seven priority-weight combinations spanning the full range of stakeholder-priority emphasis

4.1.7 Results and Interpretation

The microgrid operator first saw declining energy service. The operator could have framed the situation as insufficient energy allocation. The operator instead frames the situation as Fixes That Fail, since discharging the battery aggressively gains short-term relief while leaving less battery capability for future decisions. Readers see, in the results below, what a decision-maker gains, and gives up, by using that frame rather than an unframed allocation view. In this section, decision-makers compare the baseline policy and the CACCS policy across seven evaluated weight scenarios (S1 through S7), following the closed-loop procedure of Method Steps 6 through 8. For each weight combination, decision-makers solve the compromise Decision Support Problem for each day of the 30-day decision horizon, update the battery state variables externally, and use the updated states as the initial conditions for the next decision step. At the end of the horizon, designers compute aggregate service outcomes for each weight combination.

Designers report the baseline-policy results in Table 4.2 and the CACCS-policy results in Table 4.3. In Table 4.4, decision-makers compare the two policies directly using the average daily shortfalls for critical services, households, and productive users, together with the final battery state of health. Designers then use Figures 4.0a and 4.0b to interpret how these outcomes vary across the ternary weight space defined by w_C, w_H, w_P , where $w_C + w_H + w_P = 1$.

Table 4.2. Baseline policy results across seven weight scenarios over the 30-day evaluation period.

Sce n.	wC	wH	wP	Tot. Short (kWh)	Var. Short (kWh 2)	SO H	Short C	Short H	Short P
S1	1	0	0	3900	0	0.95 1	0	34.23	39.65
S2	0	1	0	2738.8 3	181.08	0.95 1	11.48	7.40	50
S3	0	0	1	2918.8 3	43.98	0.95 1	10.52	63.37	0
S4	0.5	0.5	0	2726.8 3	210.12	0.95 1	8.53	10.35	50
S5	0.5	0	0.5	2786.8 3	126.77	0.95 1	0	72.57	0
S6	0	0.5	0.5	2711.8 3	259.93	0.95 1	12	11.74	45.14
S7	0.33 3	0.33 3	0.33 3	2711.8 3	259.93	0.95 1	10.48	11.74	46.67

As shown in Table 4.2, the baseline policy redistributes service shortfalls across the three stakeholder groups according to the selected weight combination, while the final battery state of health remains the same in all seven scenarios at 0.951. Designers observe from this result that, over the 30-day horizon, the weight combinations change who absorbs the shortfall more than they change the total battery degradation accumulated during the horizon. When designers assign full priority to critical services in Scenario S1, designers eliminate critical-service

shortfall entirely but accept average daily shortfalls of 34.23 kWh for households and 39.65 kWh for productive users. When designers assign full priority to households in Scenario S2, household shortfall decreases to 7.40 kWh/day, while critical and productive shortfalls increase to 11.48 and 50 kWh/day, respectively. When designers assign full priority to productive use in Scenario S3, productive shortfall decreases to zero, while household shortfall rises to 63.37 kWh/day. In the mixed-weight scenarios S4 through S7, designers obtain compromise allocations that lie between these single-priority extremes.

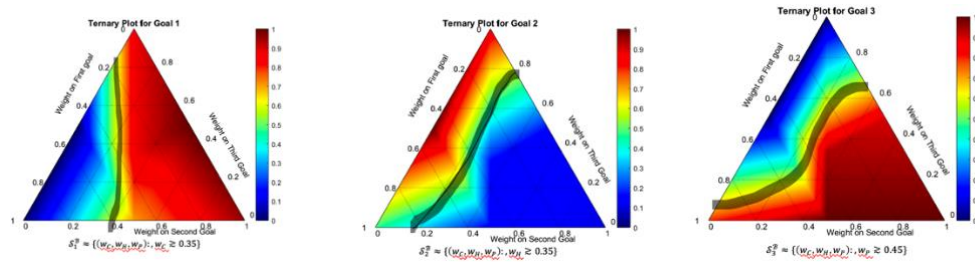


Figure 4.2. Ternary plots illustrating baseline policy energy allocation among critical services, households, and productive users across the 30-day decision horizon.

In Figure 4.2, designers map each scenario to a point in the simplex defined by $w_C + w_H + w_P = 1$. Designers place the single-priority solutions at the three vertices because one stakeholder group receives full priority at each vertex, the pairwise compromise cases along the edges because two stakeholder groups share the full weight while the third receives zero weight, and the equal-weight case in the interior because all three stakeholder groups receive the same priority. Reading Figure 4.2 together with Table 4.2, the location of each scenario in ternary space corresponds directly to the pattern of shortfalls reported numerically in the table.

Table 4.3. CACCS policy results (discharge ceiling $B = 50$ kWh/day) across seven weight scenarios over the 30-day evaluation period.

Sce n.	wC	wH	wP	Tot. Short (kWh)	Var. Short (kWh 2)	SO H	Short C	Short H	Short P
S1	1	0	0	3900	0	1.00 0	0	34.23	39.65
S2	0	1	0	2711.8 3	148.29	0.95 1	11.48	7.40	50
S3	0	0	1	2861.8 3	31.11	0.95 1	10.52	63.37	0
S4	0.5	0.5	0	2785	158.76	1.00 0	8.53	10.35	50
S5	0.5	0	0.5	2815	128.07	1.00 0	0	72.57	0
S6	0	0.5	0.5	2785	158.76	1.00 0	12	11.74	45.14
S7	0.33 3	0.33 3	0.33 3	2785	158.76	1.00 0	10.28	11.74	46.86

As shown in Table 4.3, the added discharge constraint improves battery preservation in most scenarios. In Scenarios S1, S4, S5, S6, and S7, the final battery state of health increases from 0.951 under the baseline policy to 1.000 under CACCS. In Scenarios S2 and S3, the final battery state of health remains at 0.951, but cumulative shortfall decreases relative to the baseline, from 2738.83 to 2711.83 kWh in S2 and from 2918.83 to 2861.83 kWh in S3. Across the seven scenarios,

designers compute an average final battery state of health of approximately 0.951 under the baseline policy and approximately 0.986 under the CACCS policy. Designers do not obtain this battery-preservation benefit without tradeoffs: in several mixed-weight scenarios, cumulative shortfall increases when designers restrict battery discharge more strongly. For example, cumulative shortfall increases from 2726.83 to 2785 kWh in Scenario S4 and from 2711.83 to 2785 kWh in Scenario S7, while final battery state of health increases from 0.951 to 1.000 in both cases. Designers interpret the CACCS policy as shifting the compromise toward preservation of long-term battery capability, even when that shift requires accepting somewhat larger service deficits in the short term.

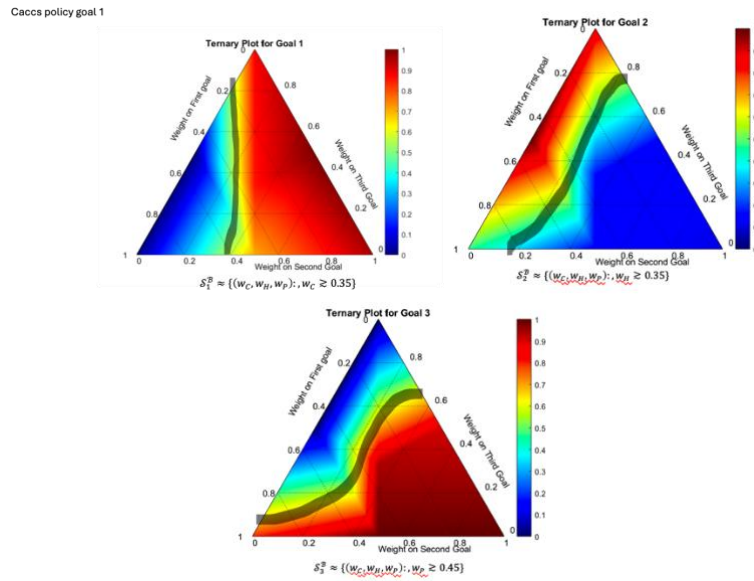


Figure 4.3. Ternary plots showing CACCS policy outcomes for energy allocation among critical services, households, and productive users across the 30-day decision horizon

When decision-makers compare Figure 4.3 with Figure 4.2, designers find that the added battery-discharge limit does not overturn the weight-driven allocation logic of the cDSP formulation. Instead, the same broad relationship between weight

selection and stakeholder allocation is preserved while the severity of battery use is moderated. Reading Figure 4.3 together with Table 4.3, CACCS changes the long-term battery outcome more than it changes the ranking of stakeholder priorities. In this sense, the ternary plots distinguish between two different effects: the effect of changing the stakeholder weights and the effect of adding the battery-preservation constraint.

Table 4.4. Comparison of baseline and CACCS policy outcomes across the seven weight scenarios.

Scen.	(wC, wH, wP)	ShortC	ShortH	ShortP	SOH (Baseline)	SOH (CACCS)
S1	(1, 0, 0)	0	34.23	39.65	0.951	1.000
S2	(0, 1, 0)	11.48	7.40	50	0.951	0.951
S3	(0, 0, 1)	10.52	63.37	0	0.951	0.951
S4	(0.5, 0.5, 0)	8.53	10.35	50	0.951	1.000
S5	(0.5, 0, 0.5)	0	72.57	0	0.951	1.000
S6	(0, 0.5, 0.5)	12	11.74	45.14	0.951	1.000
S7	(0.333, 0.333, 0.333)	10.48	11.74	46.67	0.951	1.000

As shown in Table 4.4, the stakeholder-priority structure remains largely unchanged across the two policies. When the weight shifts toward household demand, decision-makers allocate more energy toward households and accept larger shortfalls in the other demand classes; when the weight shifts toward productive use, decision-makers allocate more energy toward productive users and accept higher shortfalls in critical and household demand. These patterns remain consistent across the two policies, confirming that the weight vector continues to determine which stakeholder group receives priority. The main difference appears

in the final battery condition: under the baseline policy, the final battery state of health remains fixed at 0.951 in all seven scenarios, while under CACCS, final battery state of health reaches 1.000 in five of the seven scenarios and remains at 0.951 in the remaining two. Designers therefore conclude that the added discharge constraint changes the intensity of battery use more than it changes the allocation logic implied by the stakeholder weights.

Designers interpret these results through the Fixes That Fail archetype. Under the baseline policy, designers increase battery discharge whenever additional stored energy reduces service shortfalls, obtaining short-term relief through the balancing relief loop B1. At the same time, repeated battery discharge increases battery degradation: battery state of health decreases, usable battery capacity available for later decisions decreases, and less stored energy remains available for allocation in subsequent periods, especially during periods of low photovoltaic generation. This is the reinforcing erosion loop R1. Under the CACCS policy, designers intervene directly in this structural tension by limiting battery discharge in each decision period. This constraint weakens the short-term corrective action that provides immediate relief and therefore produces slightly larger shortfalls in some scenarios, but it slows the degradation process and preserves a larger share of usable battery capacity over the decision horizon. Designers use the CACCS formulation to make this archetypal tradeoff operational: the baseline policy leans more strongly toward short-term relief, whereas CACCS moves the compromise toward preservation of long-term capability, consistent with the structural prediction of the Fixes That Fail archetype.

4.2 Rural Water Allocation under the Tragedy of the Commons Archetype

In this section, designers apply the method developed in Chapter 3 to a rural water allocation problem. Designers follow the eight-step workflow described in Section

3.7. Designers first construct a causal loop diagram that represents the structural relationships governing groundwater use (Method Step 2). Designers then interpret the feedback structure through the Tragedy of the Commons archetype (Method Step 3) in order to identify the structural tension between individual stakeholder withdrawal and collective aquifer preservation. Designers quantify the archetypal structure (Method Step 4) and translate the result into a compromise Decision Support Problem formulation (Method Step 5). Finally, designers describe the closed-loop policy evaluation procedure (Method Steps 6 through 8) used to compare baseline and CACCS allocation policies over a thirty-day decision horizon.

4.2.1 Problem Context

Stakeholder groups first see a growing daily water shortfall. Stakeholder groups might frame the situation as insufficient allocation to whichever group is most affected. Stakeholder groups should instead frame the situation as a Tragedy of the Commons, because each group gains short-term relief by increasing its own withdrawal, while all three groups face a deeper shortfall as the shared aquifer is drawn down faster than it recharges.

Designers examine a groundwater allocation problem in a rural community that draws water from a shared aquifer. Designers consider three stakeholder groups whose activities depend on access to groundwater. The first group consists of domestic water users who require water for drinking, cooking, and basic sanitation; these uses are essential to public health and therefore carry the highest service priority. The second group consists of smallholder agricultural users who irrigate crops supporting both household consumption and local livelihoods. The third group consists of small-scale livestock and enterprise users whose activities support productive income at the village scale.

The aquifer that supplies the community recharges through seasonal rainfall and lateral inflow. On wet days, recharge raises the aquifer level. During dry periods, recharge falls and the aquifer level depends primarily on the total amount of water the three stakeholder groups withdraw. The community does not operate a centralized metering or enforcement mechanism. As a result, each stakeholder group decides independently how much water to withdraw each day in order to meet its own service requirements. Within each group, individual users tend to withdraw additional water whenever their immediate service needs increase, because each individual withdrawal has only a small effect on the shared aquifer.

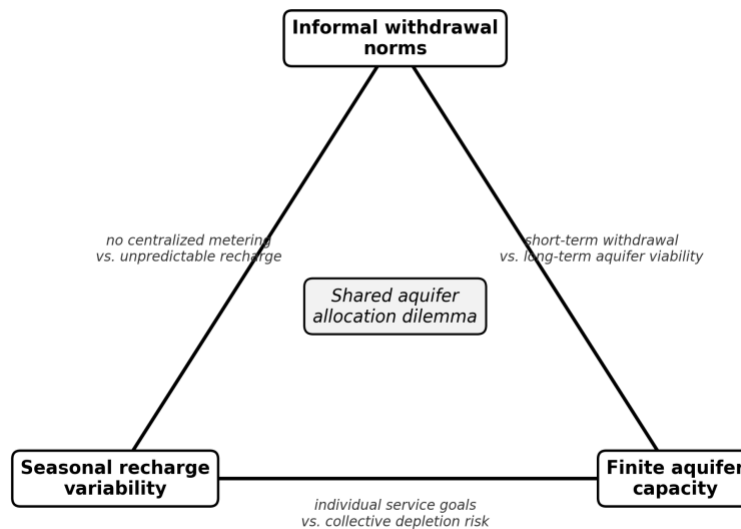
When any stakeholder group increases its withdrawal, the group reduces its own immediate service shortfall and gains short-term relief from operational pressure. However, the aquifer level reflects the sum of withdrawals across all groups. When total withdrawals exceed daily recharge, the aquifer level declines. A lower aquifer level reduces usable groundwater capacity, lengthens the time required to refill household storage, and makes it harder for every group to meet its future service requirements. As the aquifer continues to decline, each group experiences greater service pressure, which encourages further withdrawal. Through this sequence, the community shifts the structural behavior of the system from acceptable shared use toward collective resource depletion.

Through this interaction between individual withdrawal decisions and shared resource depletion, designers observe the structural feedback pattern associated with the Tragedy of the Commons archetype. In this archetype, multiple stakeholders independently pursue locally acceptable outcomes while collectively weakening a shared resource on which all stakeholders ultimately depend. In the rural water allocation problem examined in this section, each group reduces its own immediate service shortfall through additional withdrawal, but the groups' repeated withdrawal accelerates aquifer decline and increases future service shortages for

every group. This interaction creates a structural tension between two competing objectives. On one hand, each stakeholder group attempts to meet its own daily service target. On the other hand, the community as a whole attempts to preserve aquifer capability in order to maintain service continuity during future periods.

Designers design the test problem to capture this structural tension explicitly so that designers can translate the archetypal feedback structure into a quantitative decision formulation for rural water allocation. Through this test problem, designers demonstrate how decision-makers can represent the competing objectives of individual service delivery and shared aquifer preservation within a compromise Decision Support Problem framework, and how they can compare allocation policies over a thirty-day decision horizon.

Dilemma Triangle: Rural Water Allocation



Dilemma triangle (Dilemma Triangle Method, Method Step 1) for the rural water allocation problem examined in Section 4.2.

4.2.2 CLD Construction (Method Step 2)

Following Method Step 2 of Section 3.5.2, decision-makers translate the problem description into a causal loop diagram. Designers construct the CLD to represent the structural relationships among the variables that govern groundwater use in the rural community. Designers identify the key variables that capture the interaction between individual withdrawal decisions, shared aquifer level, and service delivery to the three stakeholder groups. These variables represent both the immediate corrective action taken by each group and the delayed shared-resource effects that emerge across repeated decision cycles.

The main variables represented in the CLD include:

- individual withdrawal by domestic users
- individual withdrawal by agricultural users
- individual withdrawal by enterprise users
- total community withdrawal
- aquifer recharge
- aquifer level
- usable groundwater capacity
- service shortfall for each stakeholder group
- individual pressure to withdraw

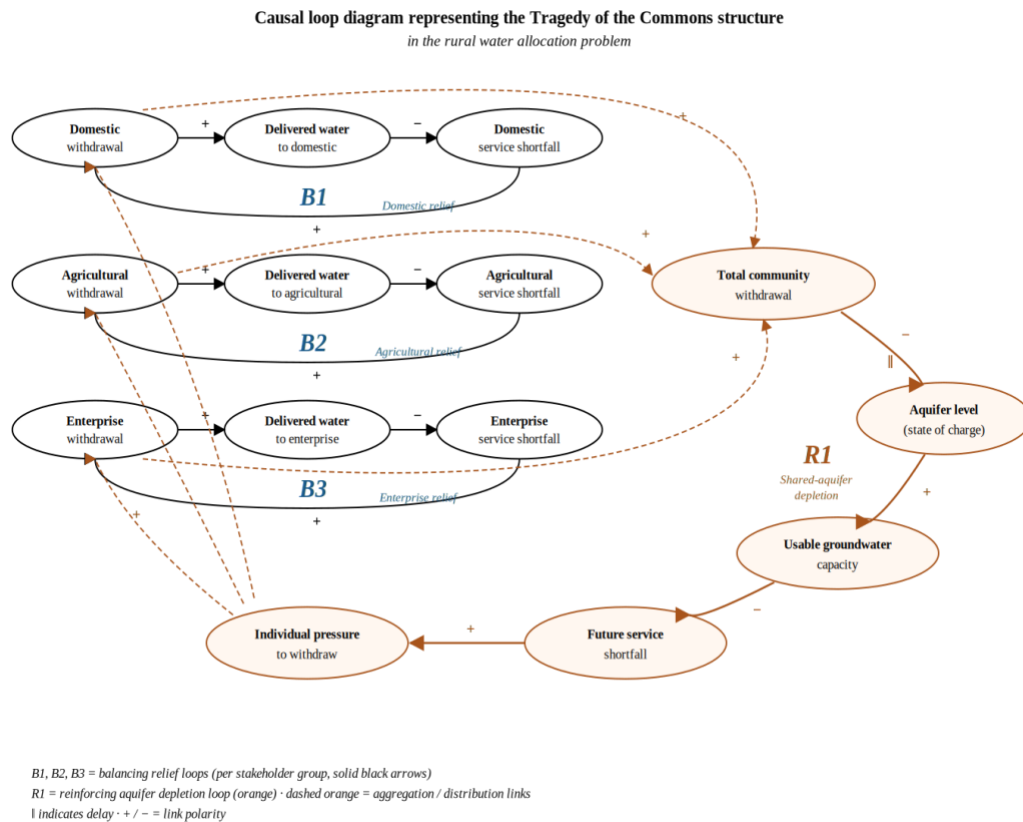


Figure 4.4. Causal loop diagram representing the Tragedy of the Commons structure in the rural water allocation problem.

In Figure 4.4, designers use these variables to establish causal relationships that describe how individual stakeholder decisions influence both immediate service delivery and shared aquifer condition over time. Through these relationships, designers construct three balancing loops associated with individual stakeholder relief and one reinforcing loop associated with collective resource erosion. These feedback structures provide the basis for the archetype interpretation described in the next step.

Designers identify three balancing loops in the diagram. Designers label these loops B1, B2, and B3 and interpret them as the immediate relief loops associated with

each stakeholder group. When the domestic group increases its withdrawal, delivered water to domestic users increases, and the domestic service shortfall decreases. The same balancing structure applies to the agricultural group (B2) and to the enterprise group (B3). Each balancing loop produces short-term service relief for the corresponding stakeholder group.

Designers also identify a reinforcing loop in the diagram and label this loop R1. When total community withdrawal increases, the aquifer level declines. As the aquifer level declines, usable groundwater capacity decreases. Reduced capacity increases the likelihood of future service shortfalls for every stakeholder group, which increases individual pressure to withdraw again. Through the interaction between the three balancing loops B1, B2, and B3 and the reinforcing loop R1, designers identify the structural feedback pattern associated with the Tragedy of the Commons archetype.

4.2.3 Archetype Interpretation (Method Step 3)

Following Method Step 3 of Section 3.5.3, designers interpret the causal loop diagram through the Tragedy of the Commons archetype. In the diagram, designers represent individual withdrawal by each stakeholder group as the corrective action that reduces immediate service shortfall for that group. When any group increases its withdrawal, delivered water to that group increases and the group's service shortfall decreases through the corresponding balancing loop. These balancing loops represent the immediate operational relief that each stakeholder group obtains when service shortages increase.

At the same time, the diagram contains the reinforcing loop R1 that represents shared resource erosion. Higher total community withdrawal lowers the aquifer level. A lower aquifer level reduces usable groundwater capacity and increases future service shortfall for every stakeholder group. Increasing future shortfall raises pressure on each group to withdraw more water in subsequent periods.

Through this sequence, the system experiences accelerating collective resource depletion that no single stakeholder controls. Through the interpretation of the three balancing loops and the shared reinforcing loop, designers identify the structural tension that defines the decision problem. Each stakeholder group seeks to reduce its own service shortfall in the present period. At the same time, the community as a whole seeks to preserve aquifer capability in order to avoid worsening shortages in future periods. In the decision formulation, designers represent this tension by defining service goals for the three stakeholder groups while designers represent aquifer preservation through the constraint set and the aquifer state update equations. Designers use this structural tension to motivate the translation from archetype interpretation to cDSP formulation in Section 4.2.4.

4.2.4 Translation to cDSP (Method Steps 4 and 5)

Following Method Steps 4 and 5 of Sections 3.5.4 and 3.5.5, decision-makers translate the structural relationships identified in the causal loop diagram into a compromise Decision Support Problem formulation. Decision-makers make the framework's context-awareness concrete for the water allocation problem through this translation: the daily aquifer state and recharge conditions sensed for that specific day (Section 3.6.1) determine the feasible withdrawal limit used in the cDSP below, so decision-makers obtain a different allocation decision from the same formulation depending on the current context, rather than applying a single allocation rule applied regardless of conditions. Designers analyze a daily water allocation problem in a rural community that draws from a shared aquifer. The community supplies water to three stakeholder groups whose activities require different levels of service reliability. Each group requires a minimum level of daily water service in order to maintain acceptable operational outcomes.

Daily aquifer recharge varies across the decision horizon. In many periods, total daily demand across the three stakeholder groups exceeds available recharge.

Under these conditions, additional water comes from stored aquifer reserves, and the aquifer level declines. When any stakeholder group increases its withdrawal, delivered water increases and the corresponding service shortfall decreases. This relationship appears in the balancing loops identified in the causal loop diagram. At the same time, repeated withdrawal across all groups lowers the aquifer level. A lower aquifer level reduces usable groundwater capacity and increases the difficulty of meeting service demand in later periods. These relationships appear in the reinforcing loop identified in the causal loop diagram.

Through the interaction of these feedback relationships, designers identify the structural pattern associated with the Tragedy of the Commons archetype. Each stakeholder group increases its withdrawal in order to reduce its own service shortfall in the present period. At the same time, repeated collective withdrawal reduces aquifer capability and increases the likelihood of future shortages for every group. The decision problem therefore contains a structural tension between short-term individual service delivery and long-term preservation of shared aquifer capability.

In the compromise Decision Support Problem formulation, designers represent this tension explicitly. Designers define daily allocation decisions for the three stakeholder groups and designers impose operational constraints that represent aquifer dynamics and feasible withdrawal limits. Through this formulation, designers determine daily allocation decisions that balance service shortfalls and shared aquifer use across the decision horizon.

In Table 4.5, designers show how key variables that describe system behavior in the qualitative feedback structure appear as decision variables, state variables, constraints, and deviation terms in the quantitative decision model.

Table 4.5. Mapping of CLD Variables to cDSP Formulation Elements (Water Allocation)

CLD Element	Role in Archetype	cDSP Component
Individual withdrawal by each stakeholder group	Corrective action (balancing loops B1, B2, B3)	Decision variable: daily allocation to each group
Delivered water to each stakeholder group	Immediate service outcome	Goal equation deviation variable
Service shortfall for each stakeholder group	Pressure that drives withdrawal	Deviation from group service target
Aquifer level	Shared state of the resource	State variable updated externally between days
Aquifer recharge	External supply to the shared resource	Given parameter for each day
Usable groundwater capacity	Long-term constraint on withdrawal	Withdrawal feasibility constraint
Total community withdrawal	Driver of reinforcing depletion (R1)	Sum constraint across stakeholder allocations
Future service shortfall and individual pressure to withdraw	Reinforcing loop R1 across cycles	Repeated solve under updated aquifer state

Given

For each day $k = 1, 2, \dots, 30$, the following are known:

- daily aquifer recharge
- daily demand of each stakeholder group (domestic, agricultural, enterprise)
- minimum acceptable daily service for each stakeholder group
- priority weights for the three service goals
- aquifer state of charge at the start of day k (updated from day $k - 1$)
- current usable groundwater capacity reflecting accumulated aquifer drawdown

Assumptions

- Decision-makers solve one static compromise Decision Support Problem for each day.
- Designers treat daily aquifer recharge as known for each day.
- Designers treat stakeholder demands as known for each day.
- Decision-makers update aquifer state variables externally after each daily solve.
- Designers use the updated aquifer state of charge as input to the next day's decision problem.
- Designers use stakeholder withdrawal as the corrective action that reduces immediate service shortfalls.
- Designers represent long-term shared-resource erosion through the aquifer feasibility constraint and aquifer state update rather than as a separate goal.

Find**System Variables**

- daily water allocation to domestic users
- daily water allocation to agricultural users
- daily water allocation to enterprise users
- daily total community withdrawal

Deviation Variables

Deviations from the daily service target for each stakeholder group:

- underachievement of the domestic service target
- underachievement of the agricultural service target
- underachievement of the enterprise service target

Satisfy**System Constraints**

- Total community withdrawal must equal the sum of the three group allocations on each day.
- Allocation to each stakeholder group must remain within its lower bound (minimum acceptable service) and upper bound (group daily demand).
- Total community withdrawal must remain within the feasible withdrawal limit determined by the current aquifer state of charge and usable groundwater capacity.
- The aquifer state of charge at the end of day k must remain nonnegative after applying recharge and subtracting total withdrawal.
- All system variables and deviation variables must remain nonnegative.

System goals

- It is desirable to achieve the target level of domestic water service.
- It is desirable to achieve the target level of agricultural water service.

- It is desirable to achieve the target level of enterprise water service.

Minimize

Designers minimize a weighted deviation function that penalizes underachievement of the three service goals across the three stakeholder groups.

Mathematical Form of the Word Problem

Designers index days by $k = 1, 2, \dots, 30$ and stakeholder groups by $i \in \{D, A, E\}$ where D denotes domestic, A denotes agricultural, and E denotes enterprise.

Given

Recharge: r_k (m^3/day , available recharge on day k)

Demands: $D_{D,k}, D_{A,k}, D_{E,k}$ (m^3/day)

Service targets: T_D, T_A, T_E

Priority weights: w_D, w_A, w_E with $w_D + w_A + w_E = 1$

Aquifer state at start of day k : S_k (m^3)

Usable capacity: C_k (m^3 , feasible drawdown for day k)

Find

System Variables

$x_{D,k}, x_{A,k}, x_{E,k}$: daily allocation to each stakeholder group (m^3)

W_k : total community withdrawal on day k (m^3)

Deviation Variables

$d_{D,k}^-, d_{A,k}^-, d_{E,k}^-$: underachievement of the daily service target for each group

Satisfy

System Constraints

1. Withdrawal balance

$$W_k = x_{D,k} + x_{A,k} + x_{E,k} \quad (4.1)$$

2. Demand bounds (each group)

$$T_i \leq x_{i,k} \leq D_{i,k} \quad \text{for } i \in \{D, A, E\} \quad (4.2)$$

3. Aquifer feasibility

$$W_k \leq C_k \quad (4.3)$$

$$S_{\{k+1\}} = S_k + r_k - W_k \geq 0 \quad (4.4)$$

4. Goal equations (deviation form)

$$x_{D,k} + d_{D,k}^- = T_D \quad (4.5)$$

$$x_{A,k} + d_{A,k}^- = T_A \quad (4.6)$$

$$x_{E,k} + d_{E,k}^- = T_E \quad (4.7)$$

5. Nonnegativity

$$x_{i,k} \geq 0, d_{i,k}^- \geq 0 \quad \text{for } i \in \{D, A, E\} \quad (4.8)$$

Minimize

$$Z_k = w_D \cdot \left(d_{D,k}^- T_D\right) + w_A \cdot \left(d_{A,k}^- T_A\right) + w_E \cdot \left(d_{E,k}^- T_E\right) \quad (4.9)$$

Decision-makers solve this compromise Decision Support Problem once for each day k . Following Method Step 7 of Section 3.7, designers then update the aquifer state of charge and usable capacity externally using equation (4.4) and a recharge update model, and designers re-solve the cDSP on day $k + 1$ using the updated state. Decision-makers use the repeated solve procedure to represent the closed-loop

operational evaluation introduced in Sections 3.4.3 and 3.6.4, exposing the structural behavior of the Tragedy of the Commons archetype across the decision horizon.

4.2.5 Results and Interpretation

Stakeholder groups drawing on the shared aquifer first saw a growing daily shortfall. Stakeholder groups could have framed the situation as insufficient allocation to whichever group was most affected. Stakeholder groups instead frame the situation as a Tragedy of the Commons, since each group's individually reasonable withdrawal collectively depletes the resource all three groups depend on. Readers see, in the results below, what decision-makers gain, and give up, by using that frame rather than an unframed allocation view. In Table 4.6, the author summarizes the 30-day outcomes across the three water allocation scenarios.

Table 4.6. Summary of 30-day outcomes across three water allocation scenarios.

Summary of Outcomes

Scenario	Weights (wD / wA / wE)	Final SoC (m ³)	Cumul. Short.D (m ³)	Cumul. Short.A (m ³)	Cumul. Short.E (m ³)
Baseline (agg = 1.30)	demand-proportional (no $w_D/w_A/w_E$ priority weights applied)	38,142	49.2	84.0	28.5
CACCS Dom Priority ($w_D=0.60$,	0.60 / 0.30 / 0.10	38,908	0	329.1	598.6

$w_A=0.30$, $w_E=0.10$					
CACCS Agri+Ent Priority ($w_D=0.20$, $w_A=0.40$, $w_E=0.40$)	0.20 / 0.40 / 0.40	40,750	2,687.7	82.0	0

In Table 4.7, the author reports the day-by-day allocations and aquifer state of charge underlying these summary outcomes.

Table 4.7. Day-by-day water allocation results. x_D , x_A , x_E in m^3/day . SoC in m^3 . R = daily recharge

Day-by-Day Results

Day	R (m^3)	Season	Baseline (agg = 1.30)				CACCS Dom Priority				CACCS Agri+Ent Priority			
			SoC	x_D	x_A	x_E	SoC	x_D	x_A	x_E	SoC	x_D	x_A	x_E
1	70	Wet	494 31	192 .5	330 .3	116	494 31	192 .5	330 .3	116	494 31	192 .5	330 .3	116
2	70	Wet	488 74	227	288 .4	111 .8	488 74	227	288 .4	111 .8	488 74	227	288 .4	111 .8
3	70	Wet	483 20	213 .9	278 .2	131 .8	483 20	213 .9	278 .2	131 .8	483 20	213 .9	278 .2	131 .8

4	70	Wet	477 06	205 .9	363 .1	114 .8	477 06	205 .9	363 .1	114 .8	477 06	205 .9	363 .1	114 .8
5	45 0	Wet	475 00	179 .4	364 .7	112 .1	475 00	179 .4	364 .7	112 .1	475 00	179 .4	364 .7	112 .1
6	45 0	Wet	473 00	179 .4	349 .6	121 .5	473 00	179 .4	349 .6	121 .5	473 00	179 .4	349 .6	121 .5
7	45 0	Wet	471 68	173 .5	301 .2	107 .1	471 68	173 .5	301 .2	107 .1	471 68	173 .5	301 .2	107 .1
8	70	Wet	466 04	222	281 .4	130 .9	466 04	222	281 .4	130 .9	466 04	222	281 .4	130 .9
9	70	Wet	460 25	206 .1	337 .7	104 .7	460 25	206 .1	337 .7	104 .7	460 25	206 .1	337 .7	104 .7
10	70	Wet	454 31	212 .5	314 .3	137 .5	454 31	212 .5	314 .3	137 .5	454 31	212 .5	314 .3	137 .5
11	45 0	Wet	452 96	171 .2	283 .7	129 .8	452 96	171 .2	283 .7	129 .8	452 96	171 .2	283 .7	129 .8
12	70	Wet	447 09	228 .2	319 .5	109 .2	447 09	228 .2	319 .5	109 .2	447 09	228 .2	319 .5	109 .2
13	70	Wet	441 82	219 .9	275 .3	102 .2	441 82	219 .9	275 .3	102 .2	441 82	219 .9	275 .3	102 .2
14	45 0	Wet	439 58	182 .7	359 .3	131 .4	439 58	182 .7	359 .3	131 .4	439 58	182 .7	359 .3	131 .4
15	45 0	Dry	438 03	180 .9	296 .8	127 .4	438 03	180 .9	296 .8	127 .4	439 34	50. 5	296 .8	127 .4

16	45 0	Dry	436 08	181 .6	335 .2	128 49	436 181	181 .6	335 9	87. 09	439 10.	10. 6	335 .6	128 .2
17	45 0	Dry	434 38	188 .3	301 .9	129 .8	434 96	188 .3	301 .9	112 .2	438 85	42. 5	301 .9	129 .8
18	70	Dry	428 80	201 .5	321 .9	104 .7	429 66	201 .5	294 .1	104 .7	434 81	47. 4	321 .9	104 .7
19	70	Dry	423 15	195 .9	324 .5	114 .9	424 43	195 .9	324 .5	72. 5	430 81	30. 2	324 .5	114 .9
20	45 0	Dry	421 82	187 .5	289 .7	106 .2	423 10	187 .5	289 .7	106 .2	430 66	69. 4	289 .7	106 .2
21	70	Dry	415 94	193 .8	340 .2	124 96	417 96	206 .7	365 .1	12. 1	426 71	0	332	133 .1
22	45 0	Dry	413 95	178 .3	346 .2	124 .4	416 69	178 .4	346 .4	52	426 60	0	336 .4	124 .4
23	45 0	Dry	411 99	182 .5	352 .4	110 .8	415 44	187 .5	273 .6	113 .9	426 49	0	346 .8	113 .9
24	70	Dry	406 26	188 .6	351 .6	102 .5	410 41	192	277	104 .3	422 59	0	356 .3	104 .3
25	70	Dry	400 62	195 .5	326 .2	112 .1	405 44	197 .4	329 .4	39. 6	418 72	13. 8	329 .4	113 .2
26	70	Dry	395 07	196 .3	325 .9	102 .8	400 55	217 .1	228 .7	113 .7	414 90	0	338 .5	113 .7
27	45 0	Dry	393 67	182 .5	280 .3	128 52	399 52	182 .5	280 3	90. 92	414 92	39. 3	280 .5	128 .3

28	70	Dry	388 23	200	289 .6	124 .4	394 71	200 .9	290 .8	59. 7	411 14	32. 3	290 .8	125
29	70	Dry	382 87	202 .1	271 .8	131 .7	389 96	205 .5	276 .3	62. 8	407 40	33. 7	276 .3	133 .9
30	45 0	Dry	381 42	172 .8	303 .2	119	389 08	172 .8	303 .2	62. 1	407 50	17. 8	303 .2	119
Final SoC (m³)			38,142				38,908				40,750			
Cumul. Short.D (m³)			49.2				0				2,687.7			
Cumul. Short.A (m³)			84.0				329.1				82.0			
Cumul. Short.E (m³)			28.5				598.6				0			

The functional difference this framing makes is visible directly in the aquifer's final state of charge: 38,142 cubic meters under the baseline, compared with 38,908 and 40,750 cubic meters under the two CACCS scenarios, a difference attributable entirely to whether the dry-season constraint driven by the Tragedy of the Commons frame was present or absent.

Baseline: Uncoordinated Extraction and Structural Depletion

The baseline policy allocates water in proportion to stochastic daily demand without imposing any conservation constraint. During the wet season (days 1-14), daily recharge of 450 m³ on high-recharge days partially offsets extraction, and the aquifer SoC declines gradually from 49,431 m³ on day 1 to 43,958 m³ on day 14.

On low-recharge days (70 m^3), the aquifer loses 550 to 700 m^3 per day net, reflecting full demand-driven extraction across all three user groups. At the dry-season transition on day 15, daily recharge falls to 70 m^3 while total demand remains in the range of 580 to 720 m^3 per day. Decision-makers following the baseline policy apply no safe yield constraint and continue allocating at the same demand-driven rate. SoC declines from $43,803 \text{ m}^3$ on day 15 to $38,142 \text{ m}^3$ on day 30, a net loss of $5,661 \text{ m}^3$ across the 16 dry-season days. Cumulative shortfalls across the 30-day horizon are modest in absolute terms - 49.2 m^3 domestic, 84.0 m^3 agricultural, 28.5 m^3 enterprise - indicating that the baseline successfully meets most daily demands by drawing down the shared resource without restraint. This is the structural behavior of the Tragedy of the Commons archetype: each user's demand is individually satisfied in the short term while the commons is collectively depleted.

CACCS Dom Priority: Domestic Protection with Aquifer Preservation

The CACCS Dom Priority scenario assigns weights $w_D = 0.60$, $w_A = 0.30$, $w_E = 0.10$, reflecting a domestic-priority orientation. During the wet season (days 1-14), all three scenarios are identical because the dry-season safe yield constraint has not yet activated. Differentiation begins at day 15. On day 15, the dry-season constraint activates. The first visible curtailment appears on day 16, where enterprise allocation drops from 128.2 m^3 (baseline) to 87.9 m^3 under Dom Priority. The domestic weight being highest means the DSIDES protects domestic supply preferentially and enterprise allocation bears the early shortfall. By day 21, enterprise allocation falls to 12.1 m^3 as aquifer conservation pressure intensifies. Cumulative enterprise shortfall over 30 days reaches 598.6 m^3 , while domestic shortfall is completely eliminated. Agricultural shortfall accumulates to 329.1 m^3 .

The final SoC under CACCS Dom Priority is $38,908 \text{ m}^3$, a modest improvement of 766 m^3 over the baseline. The aquifer is better preserved than the baseline, but the

improvement is limited because the domestic priority weight still permits relatively high domestic extraction to protect household supply. This is structurally coherent with the Tragedy of the Commons archetype: coordinated domestic protection reduces overall depletion but shifts shortfalls to lower-priority users.

CACCS Agri+Ent Priority: Aquifer Preservation through Domestic Suppression

The CACCS Agri+Ent Priority scenario assigns weights $w_D = 0.20$, $w_A = 0.40$, $w_E = 0.40$, giving equal priority to agricultural and enterprise supply. The low domestic weight drives a substantially more aggressive safe yield constraint at the dry-season transition, reflecting the policy intent of preserving water for productive use.

On day 15, the constraint activates and immediately curtails domestic allocation from 180.9 m³ (baseline) to 50.5 m³, a 72 percent reduction in a single day. From days 21 through 26, domestic allocation falls to zero across multiple consecutive days. The aquifer SoC is maintained well above the baseline trajectory throughout the dry season. The final SoC reaches 40,750 m³ on day 30, a 2,608 m³ improvement over the baseline and a 1,842 m³ improvement over CACCS Dom Priority. Agricultural shortfall is nearly eliminated (82.0 m³), and enterprise shortfall is completely eliminated. The cost of the Agri+Ent Priority policy is a cumulative domestic shortfall of 2,687.7 m³, representing a substantial sacrifice of household water supply in favor of aquifer preservation and productive-use continuity. The structural logic is consistent with the Tragedy of the Commons archetype: coordinated suppression of the dominant extractor preserves the commons for productive users and slows collective depletion. Designers present the per-scenario ternary plots for the water allocation problem in Figures 4.5, 4.6, and 4.7, and the combined acceptable-weight region in Figure 4.8.

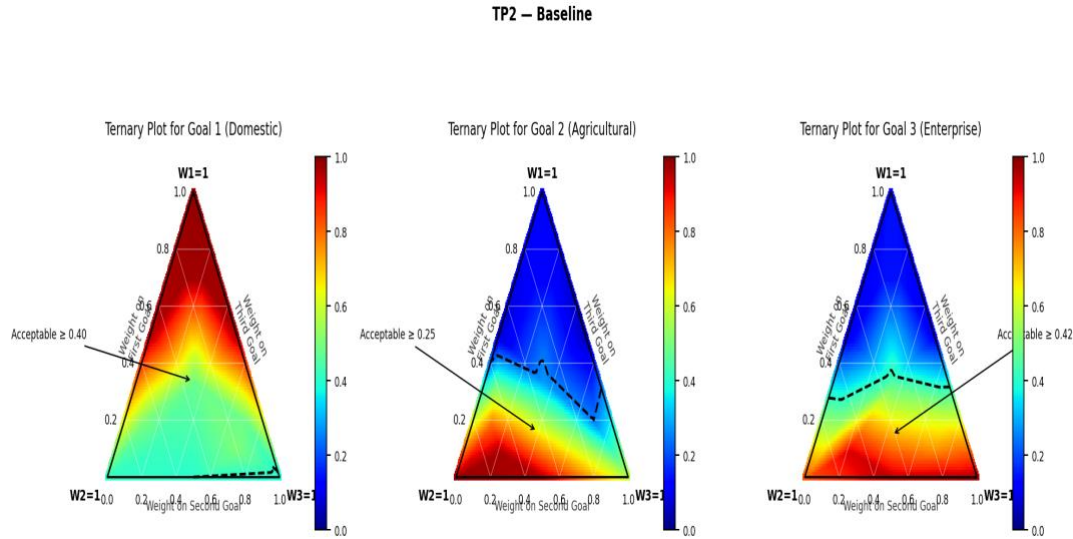


Figure 4.5. Ternary plots for TP2 Baseline (agg=1.30). Goals: Domestic, Agricultural, Enterprise.

Designers read each ternary plot as a map of the full weight space for the three service goals: each corner represents a scenario in which one goal receives all of the priority weight ($W1=1$, $W2=1$, or $W3=1$ respectively), and any point inside the triangle represents a specific combination of weights on the three goals that sums to one. The shading at each point encodes how well that weight combination satisfies the goal named in the plot title, on a 0.0-to-1.0 scale shown in the color bar; darker or lighter regions correspond to lower or higher achievement of that specific goal, and the dashed contour marks the acceptable threshold labeled on the plot (for example, "acceptable ≥ 0.40 "). Designers generate three such plots per scenario, one per goal, so that the region of weight space acceptable to all three goals simultaneously can be identified by overlaying all three acceptable regions, which designers do explicitly in the superimposed satisficing space figures that follow (Figures 4.8 and 4.13).

TP2 – CACCS Dom Priority

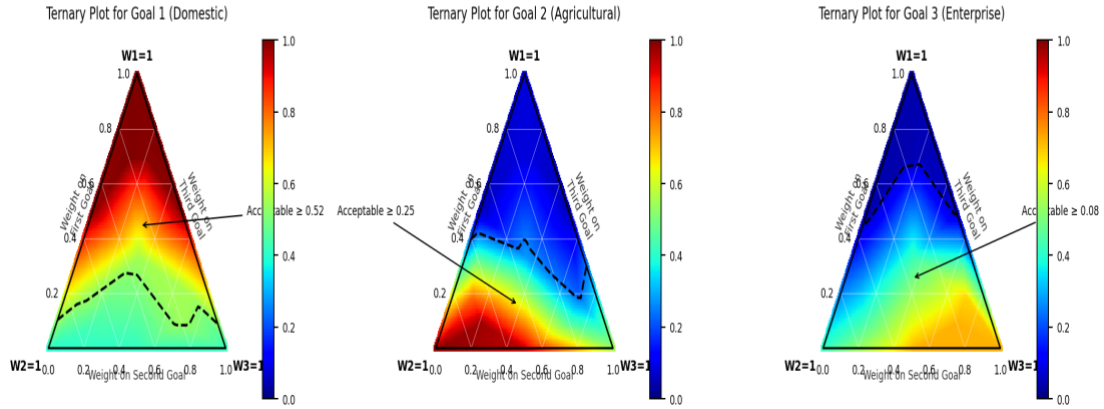


Figure 4.6. Ternary plots for TP2 CACCS Dom Priority ($w_D=0.60$, $w_A=0.30$, $w_E=0.10$).

TP2 – CACCS Agri+Ent Priority

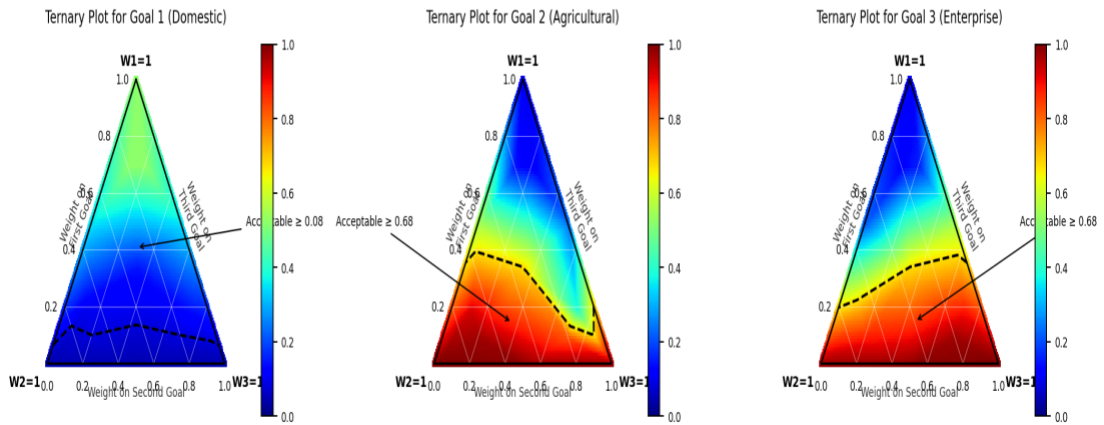


Figure 4.7. Ternary plots for TP2 CACCS Agri+Ent Priority ($w_D=0.20$, $w_A=0.40$, $w_E=0.40$).

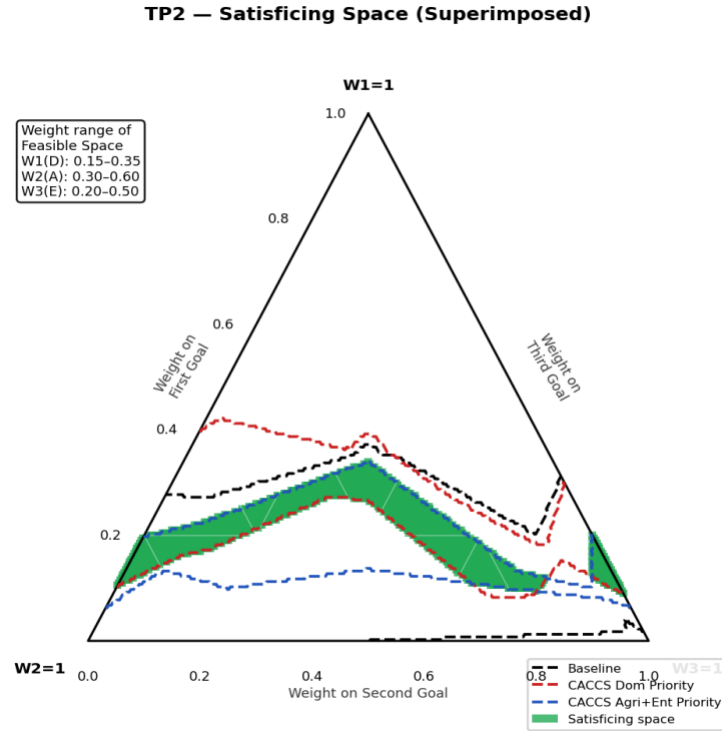


Figure 4.8. TP2 superimposed satisficing space.

This figure overlays the acceptable-region boundary from each of the three per-goal ternary plots onto a single triangle, for each of the three scenarios (baseline, CACCS Dom Priority, CACCS Agri+Ent Priority). The shaded gray region marks the weight combinations that fall inside all three scenarios' acceptable regions simultaneously, and the boxed weight ranges (for example, W1(D): 0.15-0.35) report the span of each individual weight for which this joint feasibility holds. Designers use this figure to answer a question the individual ternary plots cannot answer on their own: not merely whether a single scenario satisfies a single goal, but which range of priority weights a decision-maker could choose so that all three service goals remain acceptably met across all three policy scenarios at once. A narrow or empty overlap region indicates that the goals are in sharper structural

tension under the governing archetype, while a wide overlap indicates that a broad range of decision-maker priorities would all produce satisficing outcomes.

Cross-Scenario Comparison

The three scenarios demonstrate the structural behavior of the Tragedy of the Commons archetype and the ability of the CACCS framework to modulate it. Under the baseline policy, designers observe the archetype's characteristic outcome: individual demands satisfied in the short term while the shared resource depletes. The CACCS scenarios introduce a coordinated conservation signal through the dry-season safe yield constraint, creating a structurally informed policy that distinguishes between user priorities and enforces allocation discipline at the seasonal transition.

The key insight across all three scenarios is that aquifer preservation is achieved at a cost. Under CACCS Dom Priority, enterprise users bear the shortfall. Under CACCS Agri+Ent Priority, domestic users bear a much larger shortfall. Neither scenario eliminates shortfalls entirely. Both produce satisficing solutions that reflect the weight structure of the decision-maker's priorities. This is the behavior the cDSP formulation is designed to produce: a spectrum of weight-dependent satisficing policies that trade off user shortfalls against aquifer preservation in a structurally transparent way.

The seasonal transition at day 15 is the pivotal event across all three scenarios. Designers achieve the aquifer preservation effect in the CACCS framework entirely through the dry-season constraint, which does not activate during the wet season. This targeted activation is consistent with the context-awareness of the CACCS architecture: decision-makers use the control system to respond to changing environmental conditions rather than apply a uniform policy across all periods.

4.3 Rural Healthcare Triage under the Shifting the Burden Archetype

In this section, designers apply the method developed in Chapter 3 to a rural healthcare triage problem. Designers follow the eight-step workflow described in Section 3.7. Designers first construct a causal loop diagram that represents the clinic's operating environment, then interpret it through the Shifting the Burden archetype.

4.3.1 Problem Statement

The clinic administrator first sees a growing daily backlog of acute patients. The administrator might frame the situation as a triage problem. The administrator should instead frame the situation as Shifting the Burden, because the administrator gains short-term relief by allocating more capacity to symptomatic treatment, while the community's preventive care, and its long-term health capability, is displaced by that same allocation. Designers examine a daily clinic capacity allocation problem in a rural primary health center operating under the Shifting the Burden archetype. The clinic serves a community in which residents experience recurring health conditions arising from environmental exposure, limited sanitation, and inconsistent access to preventive services. The clinic operates with limited daily staff hours that vary stochastically around a base of 28 staff-hours per day, reflecting realistic rural conditions in which staffing fluctuates due to personnel availability, transportation access, and local health crises. The administrator allocates available clinic capacity across three competing activity categories. The first is symptomatic treatment of acute cases presenting at the clinic on the current day, including acute respiratory infections, diarrhea, fevers, and minor injuries. The second is preventive care, encompassing immunization, antenatal visits, growth monitoring, and household-level water and sanitation outreach. The third is administrative continuity, covering operational coordination, records management, and supply chain activities that the clinic must perform each day to remain

functional. Designers impose a daily minimum administrative floor M_{min} reflecting stochastic variation in mandatory administrative load ranging from 3 to 5 hours per day.

4.3.2 CLD Construction (Method Step 2)

Following Method Step 2 of Section 3.5.2, designers construct a causal loop diagram representing the feedback relationships among symptomatic treatment, preventive care, administrative continuity, community health capability, and the acute caseload presenting at the clinic each day.

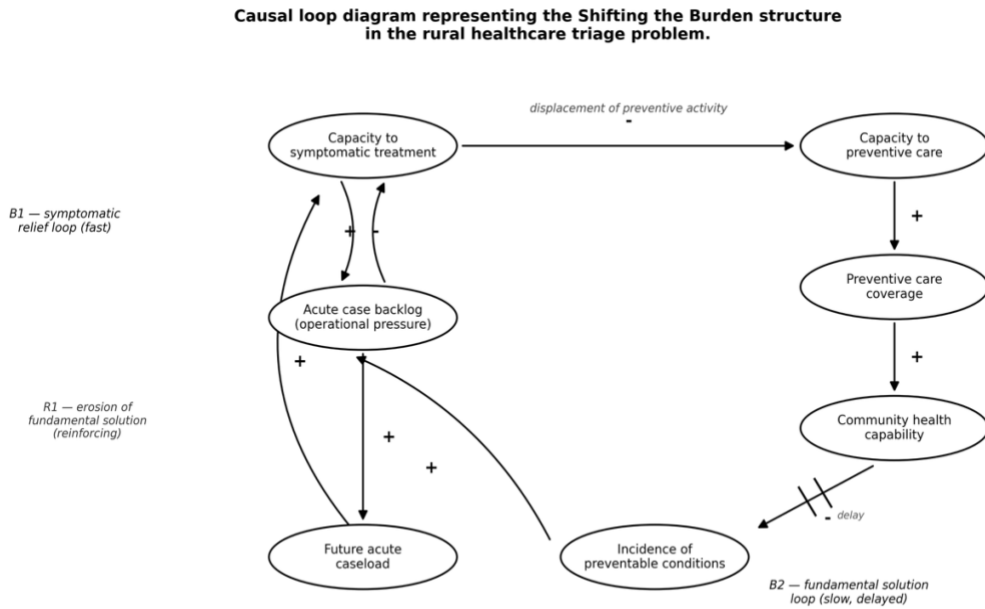


Figure 4.9. Causal loop diagram representing the Shifting the Burden structure in the rural healthcare triage problem.

4.3.3 Archetype Interpretation (Method Step 3)

The clinic operates under the Shifting the Burden archetype because, when the administrator repeatedly allocates capacity to symptomatic treatment, the administrator reduces the immediate acute backlog through the symptomatic relief loop B1 but simultaneously displaces preventive care. By displacing preventive care, the administrator weakens community health capability H^c through the reinforcing erosion loop R1. As H^c declines, the expected acute caseload A_k on subsequent days increases via the state update equation (4.18), which creates additional pressure to allocate more capacity to symptomatic treatment. Through this structural feedback, the clinic becomes progressively dependent on symptomatic response as its operational mode, which is precisely the behavior designers expect from the Shifting the Burden archetype.

4.3.4 Compromise Decision Support Problem Formulation

Given

For each day $k = 1, 2, \dots, 30$:

Parameter	Symbol	Value / Range	Source
Daily clinic capacity	H_k	$28.0 \pm 22\%$ hrs/day (stochastic)	Assumed: rural clinic staffing variability
Daily admin floor	$M_{min, k}$	U(3.0, 5.0) hrs/day	Stochastic daily admin load
Symptomatic productivity	α_s	1.5 cases/staff-hour	Assumed: rural clinic throughput

Preventive productivity	α_P	2.0 units/staff-hour	Assumed: rural outreach throughput
Admin productivity	α_M	1.0 units/staff-hour	Assumed: rural admin throughput
Symptomatic target	T_S	16.0 cases/day	Assumed: baseline community demand
Preventive target	T_P	20.0 units/day	Assumed: preventive coverage goal
Admin target	T_M	10.0 units/day	Assumed: continuity requirement
Reference caseload	A_0	20.0 cases/day	Assumed: healthy community baseline
Capability sensitivity	β	0.35	Assumed: preventive-to-caseload coupling
Initial capability	H_c^0	0.60	Assumed: moderate initial health state
Improvement rate	γ	0.0020	Assumed: slow capability recovery
Decay rate	δ	0.008	Assumed: background capability decline
Preventive threshold	$T_P, base$	10.0 units/day	Assumed: maintenance threshold
Decision horizon	N	30 days	Assumed: monthly operational cycle

Find

For each day k , find the daily staff-hour allocations:

System Variables

$h_{S,k}$ - staff-hours allocated to symptomatic treatment on day k

$h_{P,k}$ - staff-hours allocated to preventive care on day k

$h_{M,k}$ - staff-hours allocated to administrative continuity on day k

Deviation Variables

d_S^-, k - underachievement of the symptomatic-response target on day k

d_P^-, k - underachievement of the preventive-coverage target on day k

d_M^-, k - underachievement of the administrative continuity target on day k

Satisfy**Capacity balance (equality constraint):**

$$h_{S,k} + h_{P,k} + h_{M,k} = H_k$$

Administrative floor (inequality constraint):

$$h_{M,k} \geq M_{min,k}$$

Symptomatic floor (feasibility constraint):

$$h_{S,k} \geq 0.35 \times \frac{A_k}{\alpha_S}$$

The clinic must treat at least 35% of presenting acute patients regardless of weight structure. Designers set this constraint to reflect the ethical and operational minimum for a functioning rural clinic and to prevent the Shifting the Burden trap from eliminating symptomatic care entirely.

Feasibility upper bound:

$$h_{S,k} \leq \frac{A_k}{\alpha_S}$$

The clinic cannot treat more patients than present on day k .

State update equations:

$$H_{k+1}^c = H_k^c + \gamma (\alpha_P \times h_{P,k} - T_{P,base}) - \delta \quad (4.17)$$

$$A_{k+1} = A_0 (1 - \beta \times H_{k+1}^c) + \varepsilon_{k+1} \quad (4.18)$$

Community health capability H^c is clamped to $[0, 1]$ after each update. Equation (4.17) encodes the reinforcing erosion loop R1: when preventive allocation falls below $T_{P,base}$, capability declines and future acute caseload $A_{\{k+1\}}$ increases through equation (4.18).

Minimize

$$\begin{aligned} Z_k = & w_S \times \frac{(h_{S,ref} - h_{S,k})}{h_{S,ref}} + w_P \times \frac{(h_{P,ref} - h_{P,k})}{h_{P,ref}} \\ & + w_M \times \frac{(h_{M,ref} - h_{M,k})}{h_{M,ref}} \end{aligned}$$

where $h_{i,ref} = \frac{T_i}{\alpha_i}$ is the reference allocation needed to fully meet goal i . This normalized Archimedean objective maximizes weighted goal achievement across all three activity categories simultaneously.

4.3.5 Results and Interpretation

The clinic administrator first saw a growing daily backlog of acute patients. The administrator could have framed the situation as a triage problem. The administrator instead frames the situation as Shifting the Burden, since allocating capacity to relieve today's backlog displaces the preventive care that keeps tomorrow's backlog from growing. The results below show what a decision-maker gains, and gives up, by using that frame rather than an unframed allocation view.

Interpretation

Baseline: Structural Collapse of Community Health Capability

The baseline policy allocates clinic capacity using a purely reactive heuristic: 75% of available capacity after the administrative floor is directed to symptomatic treatment, with 15% of the remainder going to preventive care. This allocation produces an immediate visible response to acute presentations but generates negligible preventive coverage. The consequences are structurally predictable. On day 1, the baseline allocates $hS = 11.03$ hrs and $hP = 1.66$ hrs, producing $\alpha_P \times hP = 3.32$ preventive units against a threshold of $T_P, base = 10.0$. The state update equation (4.17) yields a negative daily change in H^c of approximately -0.008 per day. Community health capability declines from $H^c = 0.60$ on day 1 to $H^c = 0.000$ by day 30, representing complete structural collapse. As H^c declines, equation (4.18) drives A_k upward from 16.55 on day 1 to 19.56 on day 30. The clinic faces a progressively heavier acute burden despite treating every patient who presents, because the community's underlying health capability is eroding through the reinforcing loop R1. The cumulative preventive shortfall reaches 496.5 units across 30 days, confirming that the baseline policy sacrifices the fundamental solution in favor of sustained symptomatic response. This is the defining structural behavior of the Shifting the Burden archetype: short-term symptomatic relief reinforces the very conditions that increase future acute demand.

CACCS SP: Symptomatic Priority with Partial Capability Recovery

The CACCS SP policy assigns weights $wS = 0.60$, $wP = 0.25$, $wM = 0.15$, reflecting a symptomatic-priority orientation that nonetheless allocates meaningful capacity to preventive care. The cDSP satisficing engine balances these competing priorities against the stochastic daily clinic capacity H_k and the daily administrative floor $M_{min, k}$. The structural consequences differ markedly from the baseline. The DSIDES consistently allocates substantially more capacity to preventive care than the baseline: hP ranges from 7.55 hrs on day 7 to 20.54 hrs on day 12, compared to

baseline values of 0.93 to 2.86 hrs. This preventive allocation generates α_{hp} values well above $T_{p,base} = 10.0$ on most days, allowing H^c to improve from 0.60 on day 1 and reach 1.000 by day 15. Once H^c reaches its ceiling, equation (4.18) drives A_k downward to approximately 10.4-13.5 cases per day from day 15 onwards, a structural reduction of 25-35% relative to the baseline acute caseload. The cost of the SP policy is a cumulative symptomatic shortfall of 79.9 cases across 30 days, arising from days when the cDSP allocates capacity away from symptomatic treatment to protect preventive coverage. The preventive shortfall is near zero at 8.5 units. This tradeoff reflects the SP weight structure: symptomatic treatment is the priority, but preventive care receives enough weight to prevent the structural erosion that characterizes the baseline.

CACCS BA: Preventive Priority with Full Capability Restoration

The CACCS BA policy assigns weights $w_S = 0.20$, $w_P = 0.55$, $w_M = 0.25$, reflecting a preventive-priority orientation in which community health capability preservation is treated as the primary operational objective. The higher weight on preventive care drives the DSIDES to allocate substantially more capacity to hP, accepting larger symptomatic shortfalls in exchange for faster H^c recovery. The structural consequences are the most favorable of the three scenarios. Preventive allocations are consistently higher than SP: hP ranges from 15.24 hrs on day 7 to 26.43 hrs on day 12. Community health capability reaches 1.000 by day 9, six days earlier than the SP policy. Once H^c stabilizes at 1.000, A_k stabilizes at approximately 10.1-13.5 cases per day, the lowest acute caseload trajectory of all three scenarios. The preventive shortfall is zero across all 30 days. The cost of the BA policy is a cumulative symptomatic shortfall of 342.4 cases, substantially higher than SP. This reflects the BA weight structure: when preventive care must be protected, the DSIDES cuts symptomatic allocation first, accepting acute backlog in the short term in order to break the Shifting the Burden spiral earlier.

TP3 – Baseline

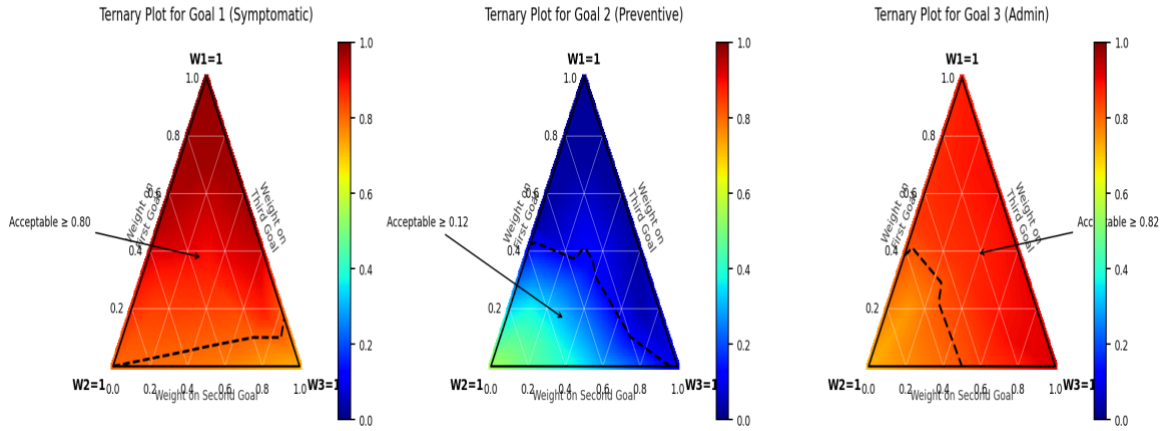


Figure 4.10. Ternary plots for TP3 Baseline. Goals: Symptomatic, Preventive, Admin.

TP3 – CACCS SP

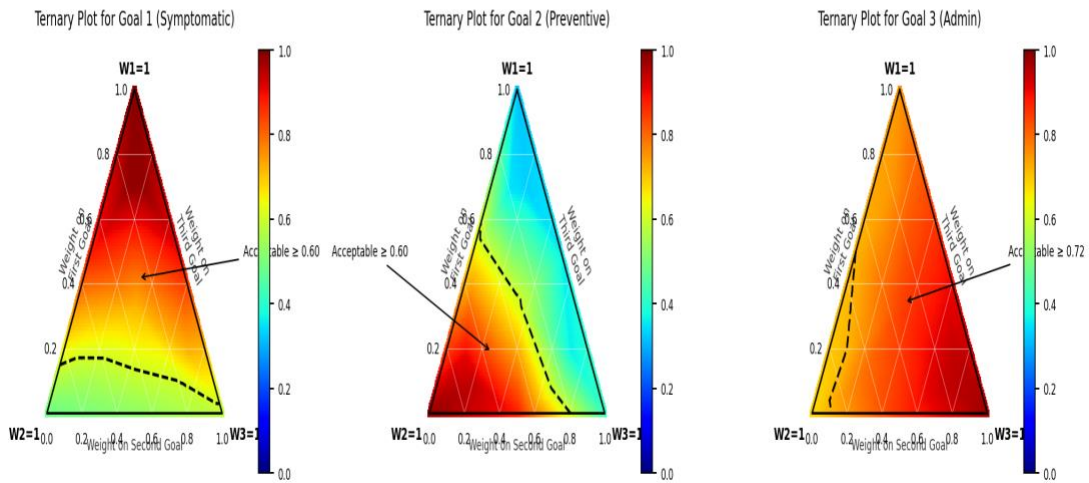


Figure 4.11. Ternary plots for TP3 CACCS SP ($wS=0.60$, $wP=0.25$, $wM=0.15$).

TP3 – CACCS BA

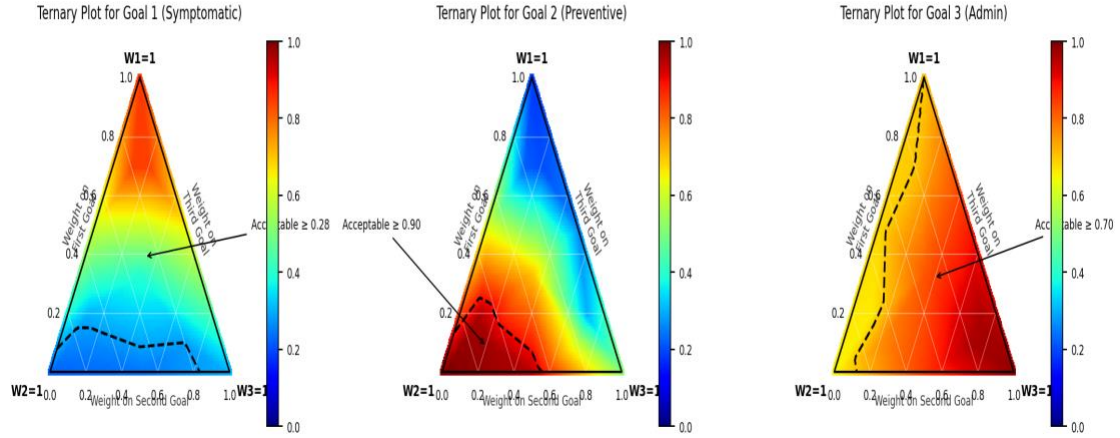


Figure 4.12. Ternary plots for TP3 CACCS BA ($wS=0.20$, $wP=0.55$, $wM=0.25$).

TP3 – Satisficing Space (Superimposed)

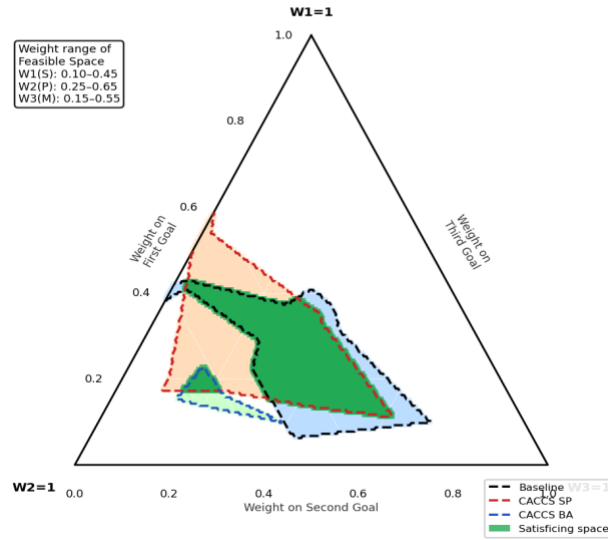


Figure 4.13. TP3 superimposed satisfying space.

As in Figure 4.10, this figure overlays the acceptable-region boundary from each per-goal ternary plot (symptomatic, preventive, administrative) across the three

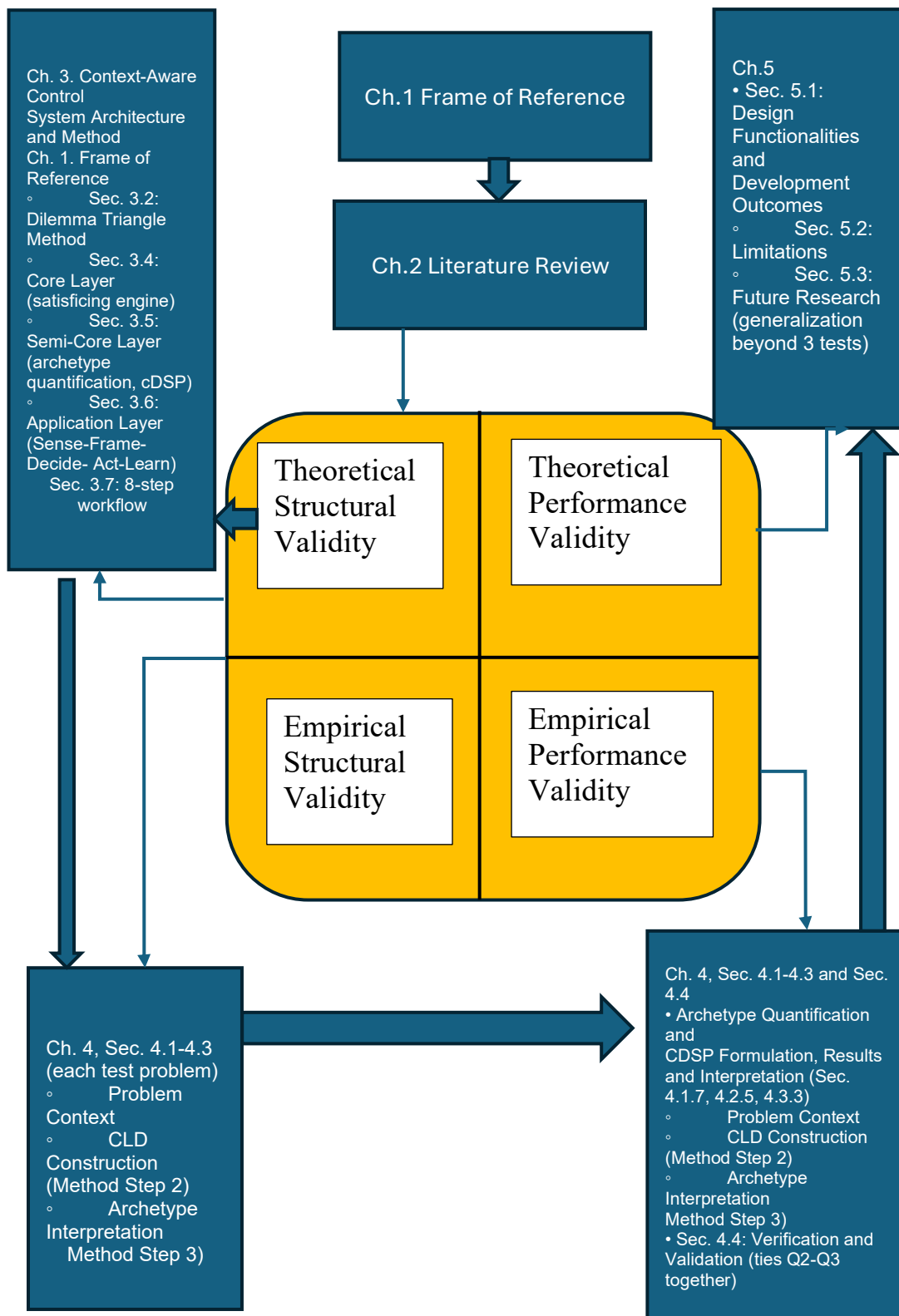
healthcare scenarios (baseline, CACCS SP, CACCS BA), and the shaded region marks the weight combinations acceptable to all three goals under all three scenarios simultaneously. Here the overlap region is visibly larger and more irregularly shaped than in the water allocation case (Figure 4.10), which reflects a structural difference between the two archetypes: Shifting the Burden trades off two loops operating on different time scales (symptomatic relief versus the fundamental preventive solution) rather than three parallel balancing loops competing for a single shared resource, so the joint-feasible weight region does not shrink to the same narrow band that a Tragedy of the Commons structure produces. Designers read this as evidence that a wider range of decision-maker priorities can sustain a structurally acceptable healthcare policy than can sustain a structurally acceptable water allocation policy, consistent with the different number and arrangement of loops each archetype involves.

Structural Comparison: The Weight Structure as Policy Instrument

The functional difference this framing makes is visible directly in community health capability: the baseline collapses this capability to 0.000 by day 30, while both CACCS scenarios restore it to 1.000 by day 9 or day 15, a difference attributable entirely to whether the preventive-care weighting driven by the Shifting the Burden frame was present or absent. The three scenarios together demonstrate that the weight structure in the CACCS cDSP formulation functions as a policy instrument that directly controls the rate of structural recovery. The baseline weight structure implicitly assigns near-zero weight to preventive care, producing structural collapse within 30 days. The SP weight structure assigns moderate weight to preventive care, producing full capability recovery by day 15. The BA weight structure assigns dominant weight to preventive care, producing full capability recovery by day 9. In all three CACCS scenarios, the allocations h_S , h_P , and h_M vary every day as a function of the stochastic daily capacity H_k , the

stochastic administrative floor $M_{min,k}$ and the state-dependent acute caseload A_k . This day-to-day variation confirms that the cDSP is solved independently for each day with the updated state, producing a genuinely closed-loop satisficing policy rather than a static allocation rule. Through these results, designers confirm the **Shifting the Burden** structural prediction: any policy that chronically under-allocates to preventive care will experience accelerating acute caseload growth through the reinforcing loop R1. Designers break this spiral in the CACCS framework by embedding the preventive-coverage goal and the state update equations directly into the cDSP formulation, ensuring that the fundamental solution remains structurally protected across the decision horizon.

4.4 Verification and Validation



4.4.1 Structure of the Validation Process

Designers use the Validation Square (Pedersen, Emblemssvåg, Bailey, Allen, and Mistree, 2000) to organize evidence that decision-makers and designers satisfy the five design functionalities of Section 1.8, not to claim broad validation of a general CACCS theory. Designers use the three test problems of this chapter to provide evidence that decision-makers can frame, translate, formulate, execute, and update decisions in three rural service domains. This section does not claim that CACCS produces better outcomes in general; it organizes, functionality by functionality, the specific evidence each test problem provides. Two checks ask whether the method is internally sound: whether its individual constructs make sense on their own, and whether the constructs work correctly together. Two further checks ask whether the method is useful in practice: whether the examples chosen to test it are appropriate, and whether the results those examples produce are actually valuable. Designers address the first two checks in Chapter 3, where designers develop the method itself, and the second two in Chapter 4, where designers test it against three rural test problems. A reader can think of the Validation Square as a four-part checklist that keeps a new design method from being judged only by whether its examples happen to work, without also asking whether the method that produced them is logically sound in the first place.

Designers evaluate the Context-Aware Control System framework through a combined verification and validation process following the Validation Square structure of Pedersen, Emblemssvåg, Bailey, Allen, and Mistree (2000). Designers use the Validation Square to organize evaluation into four quadrants: Theoretical Structural Validity (Q1), Empirical Structural Validity (Q2), Empirical Performance Validity (Q3), and Theoretical Performance Validity (Q4). Designers address each quadrant explicitly across Chapters 3 and 4.

Quadrant 1 (Theoretical Structural Validity) was addressed in Chapter 3. Designers demonstrated the internal consistency of the three-layer CACCS architecture by showing that the Core Layer satisficing philosophy, the Semi-Core Layer translation procedure, and the Application Layer Sense-Frame-Decide-Act-Learn workflow form a coherent operational structure in which each layer directly supports the design functionalities specified in Chapter 1. The eight-step workflow of Section 3.7 provides a replicable step-by-step procedure that connects qualitative systems interpretation to executable satisficing decisions without logical discontinuity.

4.4.2 Empirical Structural Validity (Quadrant 2)

Quadrant 2 addresses whether the chosen test problems are structurally appropriate vehicles for testing the elements of the method. Designers argue that the three test problems - photovoltaic battery microgrid, rural water allocation, and rural healthcare triage - satisfy this requirement for three reasons.

First, each test problem is governed by a distinct systems archetype. The microgrid operates under the Fixes That Fail archetype, the water allocation problem operates under the Tragedy of the Commons archetype, and the healthcare triage problem operates under the Shifting the Burden archetype. These three archetypes differ structurally: Fixes That Fail involves a single delayed reinforcing degradation loop; Tragedy of the Commons involves multiple simultaneous balancing loops interacting with a single shared reinforcing depletion loop; Shifting the Burden involves two balancing loops operating on different time scales with a reinforcing erosion loop that displaces the long-term solution. Testing the framework across three structurally distinct archetypes provides evidence that the translation method is not specific to a single feedback structure.

Second, each test problem operates in a distinct rural service domain. The microgrid problem is in the energy domain, the water allocation problem is in the water

domain, and the healthcare triage problem is in the healthcare domain. Testing across three domains provides evidence that the framework is not specific to a single operational environment. The same Core Layer engine, the same Semi-Core Layer translation procedure, and the same Application Layer evaluation process apply across all three domains, with only the domain-specific decision variables, constraints, and goal equations differing.

Third, each test problem involves a meaningful comparison between an archetype-unaware baseline policy and archetype-aware CACCS satisficing policies. The baseline policies in all three test problems produce behavior that is structurally consistent with the governing archetype's predicted failure mode - degradation amplification in TP1, collective aquifer depletion in TP2, and community health capability collapse in TP3. This structural correspondence provides evidence that the test problems are correctly specified as vehicles for demonstrating archetypal behavior.

4.4.3 Empirical Performance Validity (Quadrant 3)

Quadrant 3 addresses whether each test problem provides evidence that decision-makers satisfy F3 and F4, formulating a satisficing decision within the frame and updating it as conditions change, rather than whether CACCS is superior in general. In each test problem, the same cDSP formulation produces a different result depending on whether the archetype-driven structural constraint is present (CACCS) or absent (baseline); this difference is evidence of functionality satisfaction in that specific test problem, not a general claim about CACCS performance.

First, designers show that the framework has the functional capacity to protect the structural state variable that designers identify, through the recognized archetype, as being at risk. In TP1, under the CACCS policy, designers obtain a higher battery

state of health at the end of the 30-day horizon than under the baseline, showing that by disciplining discharge, the operator preserves usable capacity. In TP2, under both CACCS scenarios, designers obtain higher final aquifer SoC than under the baseline ($38,908 \text{ m}^3$ and $40,750 \text{ m}^3$ versus $38,142 \text{ m}^3$), showing that the dry-season safe yield constraint lets the community preserve shared groundwater capacity. In TP3, under both CACCS scenarios, H^c reaches 1.000 by day 9 or day 15, while under the baseline, H^c falls to 0.000 by day 30, showing that by weighting preventive care sufficiently, the administrator breaks the structural erosion spiral.

Second, designers show that the framework has the functional capacity to redirect shortfall in a direction traceable to the decision-maker's stated priorities rather than to an unframed allocation rule. In TP2, under Dom Priority, designers eliminate domestic shortfall while concentrating shortfall on enterprise (598.6 m^3), and under Agri+Ent Priority, designers eliminate enterprise shortfall while concentrating shortfall on domestic ($2,687.7 \text{ m}^3$). In TP3, under SP, designers concentrate shortfall on symptomatic treatment (79.9 cases) while nearly eliminating preventive shortfall (8.5 units), and under BA, designers eliminate preventive shortfall entirely while concentrating shortfall on symptomatic treatment (342.4 cases). From these patterns, designers confirm that the weight structure functions as a structural policy instrument rather than a purely numerical parameter.

Third, designers show that the framework has the functional capacity to respond to a changing system state within a single, unmodified cDSP formulation, rather than applying a fixed rule regardless of context. In TP2, once the dry-season constraint activates at day 15, designers obtain structurally different allocation behavior from days 15 to 30 than from days 1 to 14. In TP3, the daily variation in hS,k , hP,k , and hM,k across 30 days reflects the stochastic daily capacity H_k , the stochastic administrative floor $M_{min,k}$, and the state-dependent caseload A_k . From these

variations, designers confirm that the DSIDES operates on a genuinely updated state at each decision cycle rather than applying a static rule.

Together, these three observations show that designers can use the framework to carry a recognized frame through into an operational decision that designers cannot produce through a genuinely unframed process. This is a claim about functional capability, not a claim that the specific numerical outcomes reported here are optimal or unique. In Table 4.8, the author records the Validation Square status for the CACCS framework.

4.4.4 Summary of Validation Evidence

Table 4.8. Validation Square status for the CACCS framework.

Quadrant	Type	Location	Status
Q1	Theoretical Structural Validity	Chapter 3, Sections 3.4- 3.7	Addressed: 3-layer architecture is internally consistent; 8-step workflow is replicable across domains
Q2	Empirical Structural Validity	Chapter 4, Sections 4.1- 4.3	Addressed: 3 structurally distinct archetypes $\times$ 3 rural domains; baseline behavior confirms archetypal structure
Q3	Empirical Performance Validity	Chapter 4, Sections 4.2.5 and 4.3.5	Addressed: CACCS policies preserve structural state and shift shortfalls in weight-consistent direction across all 3 TPs

Q4	Theoretical Performance Validity	Chapter 5, Section 5.1	Addressed in closing remarks: domain independence across 3 domains supports claim of generalizability
-----------	--	---------------------------	--

4.5 What Has Been Presented in Chapter 4 and What Is Next

In Chapter 4, the author asks whether decision makers who use the Dilemma Triangle Method and archetype interpretation to frame the situation first reach a materially different decision than decision makers who skip framing and optimize directly against the immediate symptom. The baseline policy in each test problem represents the unframed case: an operator who treats the situation as an ordinary allocation problem, without recognizing the governing archetype. Decision-makers use the CACCS policy to represent the framed case. The gap between them, preserved battery health in the microgrid, preserved aquifer capacity in the water allocation problem, preserved community health capability in the healthcare triage problem, is the empirical evidence that framing the situation correctly changes the decision that follows.

In this chapter, designers applied the eight-step CACCS workflow to three rural test problems. Designers constructed causal loop diagrams, interpreted feedback structures through systems archetypes, quantified archetypal behavior through CLD-to-cDSP mapping tables, and formulated daily compromise Decision Support Problems with explicit state update equations. Designers solved each cDSP across a 30-day decision horizon, updated state variables externally between days, and compared archetype-aware CACCS policies against archetype-unaware baseline policies.

The author uses the three test problems to show that decision makers apply the same CACCS method structure across distinct archetypal feedback structures and rural service domains. In Chapter 5 the author revisits the design functionalities specified in Chapter 1, states the development outcomes explicitly, acknowledges the limitations of the framework, and identifies directions for future research.

CHAPTER 5

CLOSING REMARKS

In this thesis, designers contribute a problem-framing approach for rural cyber-physical-social systems. Designers show how a decision-maker moves from a visible rural service symptom to a structural problem frame before formulating a satisficing operational decision. Designers use archetype quantification as the method for translating that frame into decision variables, constraints, weighted trade-offs, and state updates. Designers use the compromise Decision Support Problem to support a satisficing action within the frame. Through three test problems, a photovoltaic battery microgrid under the Fixes That Fail archetype, a groundwater allocation system under the Tragedy of the Commons archetype, and a primary healthcare clinic under the Shifting the Burden archetype, designers show how a decision-maker reaches a different, and more structurally appropriate, decision by framing the situation before deciding.

5.1 Design Functionalities and Development Outcomes

5.1.1 F1: Problem Framing Functionality

F1: Decision-makers need to recognize the structural tension in a rural cyber-physical-social situation before they formulate the decision problem.

Decision-makers satisfy this functionality using the three-layer CACCS architecture developed in Chapter 3. Decision-makers use the Dilemma Triangle Method to establish the drivers and tensions that characterize a rural problem space. Decision-makers use causal loop diagram construction to identify the operational variables and feedback relationships governing long-term system behavior. Decision-makers use systems archetype interpretation to name the recurring structural pattern, Fixes That Fail, Tragedy of the Commons, or Shifting the

Burden, that decision-makers identify through the causal loop diagram. Decision-makers apply this same sequence, corresponding to the Sense and Frame stages of the Application Layer (Sections 3.6.1 and 3.6.2), unchanged across all three test problems in Chapter 4, with only the domain-specific variables differing. Designers use this consistency as evidence that the same framing procedure supports context-specific decision making without requiring a separate control system for each domain.

Development Outcome: Decision-makers move from a visible service symptom to a named structural frame before they formulate the decision problem.

5.1.2 F2: Translation Functionality

F2: Decision-makers need to translate a qualitative frame, expressed through causal loop diagrams and systems archetypes, into cDSP decision variables, constraints, deviation terms, weights, and state updates.

Decision-makers satisfy this functionality using the archetype quantification method introduced in Section 3.5.4 and the CLD-to-cDSP translation procedure of Section 3.5.5. Decision-makers use this method to translate balancing loops into service-target goal equations, reinforcing loops into degradation penalties and state-update constraints, structural tensions into weighted trade-offs, and delayed effects into closed-loop state dependencies. Decision-makers apply this procedure consistently across three structurally distinct archetypes and three rural service domains, as designers document in the CLD-to-cDSP mapping tables of Chapter 4, demonstrating its replicability.

Development Outcome: Decision-makers use archetype quantification to translate a named frame into operational decision constructs.

5.1.3 F3: Satisficing Decision Functionality

F3: Decision-makers need to formulate satisficing operational decisions that account for uncertainty, limited resources, local knowledge, and stakeholder priorities.

Decision-makers satisfy this functionality using the compromise Decision Support Problem and the Core Layer satisficing engine developed in Chapter 3. Decision-makers use the cDSP formulation to accommodate multiple, partially conflicting goals through weighted deviation terms rather than a single aggregated objective, representing stakeholder priorities directly as weights. Decision-makers formulate and solve this problem during the Decide stage of the Application Layer (Section 3.6.3), using Given-Find-Satisfy-Minimize blocks built for each test problem in Chapter 4, so that decision-makers reach a satisficing decision consistent with whichever frame the situation was given in Section 5.1.1. Development Outcome: Decision-makers use the compromise Decision Support Problem to make satisficing decisions within the frame.

5.1.4 F4: Closed-Loop Execution Functionality

F4: Decision-makers need to update state variables and reformulate decisions as rural operating conditions change.

Decision-makers satisfy this functionality using the Act and Learn stages of the Application Layer's closed-loop state update procedure, described in Sections 3.6.4, 3.6.5, and 3.4.3. Decision-makers solve each day's cDSP using the currently active state variables, then update those state variables externally and reformulate the next day's cDSP using the updated state. Decision-makers repeat this cycle to carry structural memory across decision periods. Designers compare policy behavior under the stochastic operating conditions of all three test problems, variable solar

generation in TP1, stochastic demand and seasonal recharge transitions in TP2, and stochastic clinic capacity in TP3, as evidence that decision-makers can execute this update procedure under realistic uncertainty rather than only under idealized conditions, without continuous connectivity or computationally intensive infrastructure.

Development Outcome: Decision-makers use repeated state updates to examine future operating conditions after each decision.

5.1.5 F5: Cross-Domain Demonstration Functionality

F5: Designers need to show that decision-makers can use the same method structure for energy, water, and healthcare test problems, without redesigning the method for each domain.

Designers satisfy this functionality by applying the eight-step workflow of Section 3.7, unmodified, to three test problems spanning three rural service domains in Chapter 4. Designers use this comparison as evidence of functionality satisfaction in the three domains tested, not as a general claim that CACCS produces better outcomes across rural cyber-physical-social systems broadly.

Development Outcome: Designers apply the same method structure to three rural service domains.

5.1.6 Synthesis Across the Three Test Problems

Across the three test problems, designers compare baseline and CACCS policies and observe the structural failure modes that researchers associate with the governing archetypes: battery state of health erodes under Fixes That Fail, stakeholders deplete the shared aquifer under Tragedy of the Commons, and the administrator allows community health capability to fall to 0.000 by day 30 under

Shifting the Burden. In each case, designers use the CACCS policy to protect the structural state variable that decision makers identify as being at risk. Designers calculate an average final battery SOH increase from 0.951 to 0.986, calculate a final aquifer storage increase from 38,142 to as much as 40,750 cubic meters, and observe community health capability recover to 1.000 within 9 to 15 days. Designers obtain these results without changing the architecture, engine, or translation procedure for each domain. The author uses this consistency across three structurally distinct archetypes, three rural service domains, and one unmodified method as the empirical basis for the central claim: decision makers who frame the structural tension before formulating a decision reach materially different decisions than decision makers who skip framing and act only on the immediate symptom.

5.1.7 Assumptions Underlying the Framework and Its Demonstration

Before stating the limitations of the framework, the author states explicitly the assumptions underlying its development and demonstration in Chapters 3 and 4, since the limitations that follow are best understood as consequences of these assumptions.

- Single governing archetype per test problem. Each test problem in Chapter 4 is constructed so that one archetype dominates its structural behavior; the author assumes this dominant structure is separable from other, weaker feedback effects that may also be present in a real rural system.
- Parameter values are structurally motivated, not empirically fitted. Stochastic distributions, productivity rates, and state-update coefficients in Chapter 4 (Tables 4.0-4.2 and the healthcare cDSP parameter table) are chosen to produce realistic operating ranges and to exercise the governing

archetype clearly. The author assumes these values are illustrative rather than validated against field data from a specific rural community.

- Fixed weights within a decision horizon. Within each 30-day scenario, decision-makers hold the priority weights (w_D, w_A, w_E in TP2; w_S, w_P, w_M in TP3) constant. The author assumes a decision-maker's priorities do not need to change within a single operational cycle, and treats between-scenario weight changes, not within-scenario adaptation, as the mechanism for representing different policy stances.
- Perfect same-day state information. The author assumes that decision-makers have access, at the Sense stage of each daily cycle, to the true current state (battery state of charge, aquifer level, or community health capability) at the start of that day, rather than a noisy or delayed estimate of it, even though the author builds this framework for a rural environment with incomplete and delayed information.
- Independent daily solves. The author assumes decision-makers can solve each day's cDSP as a static problem given the externally updated state, rather than as a multi-period optimization that looks ahead across the full decision horizon; this is consistent with the satisficing philosophy of Section 2.4.1 but means the policies decision-makers compute in Chapter 4 account only for the current day, not for future state trajectories.
- Established causal reading. The author adopts the causal relationships documented in existing systems archetype and system dynamics literature as the basis for each causal loop diagram, rather than independently establishing causality for the specific rural systems examined in Chapter 4.

5.2 Limitations of the Framework

In this thesis, the author identifies several limitations that motivate future research.

First, the archetype quantification method as currently formulated handles single-archetype test problems. Rural systems frequently operate under simultaneously active archetypes - for example, a water system exhibiting both Tragedy of the Commons behavior in shared resource extraction and Limits to Growth behavior in community agricultural expansion. In the current framework, the author does not provide a formalized method for combining multiple archetype structures within a single cDSP formulation, although designers can use the modular Semi-Core Layer to pursue such extensions.

Second, the test problems in Chapter 4 use assumed parameter values for stochastic distributions, productivity rates, and state-update coefficients. These values are structurally motivated and calibrated to produce realistic operating ranges, but they are not derived from empirical field data from specific rural communities. Designers use the structural behavior observed across the test problems to illustrate operational logic, not to predict any particular real-world system.

Third, the current framework does not include a mechanism for adapting weights across decision cycles in response to accumulated operational experience. The weights in each CACCS scenario are fixed for the 30-day horizon. In practice, a decision-maker might adjust priorities as seasonal conditions change, as community needs evolve, or as the structural state of the system improves or deteriorates. Designers can add weight adaptation logic to help decision makers respond to long-term structural change.

Fourth, designers use the ternary satisficing space visualization to interpret weight-dependent goal achievement across three goals. Rural operational problems frequently involve more than three competing objectives. Extending the satisficing space analysis to higher-dimensional settings would require the Interpretable Self-Organizing Map approach referenced in Section 5.3.

Fifth, the physical processes underlying each test problem, battery capacity fade, aquifer recharge, and health capability decline, are nonlinear in the real systems these test problems represent. In Chapter 4, the author approximates these processes with linear relationships. For example, in Equations 7 and 8 in Section 4.1.6, the author scales usable battery capacity linearly with state of health. The author does not capture threshold effects or accelerating degradation because the author uses a simplified linear approximation. In Section 1.1, the author states this scope decision at the outset.

5.3 Future Research Directions

In this thesis, designers establish a foundation for several directions of future research.

First, designers should extend the framework to combined-archetype test problems in which two or more archetypes interact within a single rural system. A fourth test problem examining irrigation and crop protection under the Limits to Growth archetype is identified as the immediate next step. Beyond this, combined formulations in which Tragedy of the Commons and Limits to Growth archetypes interact within a shared water-energy system would extend the method to structurally richer rural environments.

Second, researchers should validate archetype quantification against empirical field data from rural communities where community members and researchers can independently identify the governing archetype through engagement and historical operational records. This empirical validation would move the framework from the second and third quadrants of the Validation Square toward the fourth quadrant of theoretical performance validity at the domain level.

Third, designers should extend the framework to incorporate weight adaptation mechanisms that allow decision-maker priorities to evolve across operational

cycles in response to changing structural conditions. Designers can use this extension to help decision makers pursue long-term adaptive management in rural environments where community priorities shift seasonally or in response to environmental shocks.

Fourth, designers should extend the ternary satisficing space visualization to higher-dimensional weight spaces through the Interpretable Self-Organizing Map approach. This extension would allow the framework to support satisficing policy evaluation for rural problems involving four or more competing service objectives.

Fifth, designers should implement the framework in a lightweight software tool suitable for use in low-connectivity rural environments. Field practitioners could use such a tool to specify archetype structures, formulate cDSP problems, and evaluate satisficing policies without relying on computational infrastructure beyond a basic computing device. This implementation work would bridge the research framework to practical rural operational decision support.

5.4 Closing Statement

In this thesis, the author develops a problem-framing method for rural cyber-physical-social systems. Decision-makers frame the rural service problem before they decide what to do about it. Decision-makers name the structural tension through a systems archetype. Decision-makers use archetype quantification to translate the frame into operational decision constructs. Decision-makers use the compromise Decision Support Problem to formulate satisficing actions within the frame. Operators execute these actions and update state variables as conditions change. Designers use the three test problems in energy, water, and healthcare to show that decision-makers can frame, translate, formulate, execute, and update decisions within the specified test problems. The principal contribution of this thesis is a problem-framing approach for rural cyber-physical-social systems, not the archetype quantification method by itself: archetype recognition and archetype

quantification make the framing operational, and the compromise Decision Support Problem translates that framing into satisficing decisions. Designers fill a gap that has persisted in the systems thinking and rural CPSS literature, where archetype identification has remained qualitative and decision formulation has proceeded independently of structural reasoning. The author establishes a foundation for this kind of problem framing, demonstrated within the three specified test problems, without claiming field validation, general CACCS superiority, or performance improvement across rural CPSS broadly.

REFERENCES

- Bertus, C., Khosrojerdi, A., Panchal, J. H., Allen, J. K., and Mistree, F., 2012, "Identifying Dilemmas Embodied in the Sustainable Development of Rural Communities," ASME Design Automation Conference, Chicago, IL, Paper Number DETC2012-71163.
- Bhalerao, M. J., Honeycutt, W. T., Allen, J. K., and Mistree, F., 2023, "Framing Wicked Problems Through Evidentiary and Interpretative Analysis," ASME Design Automation Conference, Boston, MA, Paper Number DETC2023-117285.
- Braun, W., 2002, The System Archetypes, System Dynamics Group, University of New Hampshire, Durham, NH.
- Forrester, J. W., 1961, Industrial Dynamics, MIT Press, Cambridge, MA.
- Guo, L., Allen, J. K., and Mistree, F., 2024, "Optimize or Satisfice in Engineering Design?" Research in Engineering Design, <https://doi.org/10.1007/s00163-023-00431-5>.
- Kim, D. H., 1994, Systems Archetypes II: Using Systems Archetypes to Take Effective Action, Pegasus Communications, Cambridge, MA.
- Libey, A., Chintalapati, P., Kathuni, S., Amadei, B., and Thomas, E., 2022, "Turn Up the Dial: System Dynamics Modeling of Resource Allocations toward Rural Water Supply Maintenance in East Africa," Journal of Environmental Engineering, 148(4).
- Mandegari, S., Bhalerao, M. J., Allen, J. K., and Mistree, F., 2024, "Predicting Future States of Evolving Systems," Proceedings of ASME IDETC-CIE 2024, Washington, DC, Paper Number DETC2024-146340.

Mistree, F., Hughes, O., and Bras, B., 1993, "The Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm," *Structural Optimization: Status and Promise*, AIAA, Washington, DC, pp. 247-286.

Ning, H., and Liu, H., 2015, "Cyber-Physical-Social Based Security Architecture for Future Internet of Things," *Advances in Internet of Things*, 2(1), pp. 1-7.

Pedersen, K., Emblemstvag, J., Bailey, R., Allen, J. K., and Mistree, F., 2000, "Validating Design Methods and Research: The Validation Square," *ASME Design Engineering Technical Conferences*, Baltimore, MD, Paper Number DETC2000/DTM-14579.

Rittel, H., and Webber, M., 1973, "Dilemmas in a General Theory of Planning," *Policy Sciences*, 4(2), pp. 155-169.

Sani Mohammed, S., Schaefer, D., and Milisavljevic-Syed, J., 2023, "Towards Pre-Emptive Resilience in Military Supply Chains: A Compromise Decision Support Model-Based Approach," *International Journal of Logistics Research and Applications*, <https://doi.org/10.1080/21693277.2023.2220768>.

Schön, D., 1983, *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York.

Senge, P. M., 1990, *The Fifth Discipline: The Art and Practice of the Learning Organization*, Doubleday, New York.

Sharma, G., Allen, J. K., and Mistree, F., 2023, "Designing Concurrently and Hierarchically Coupled Systems," *Engineering Optimization*, 55(9), pp. 1556-1576.

Simon, H. A., 1956, "Rational Choice and the Structure of the Environment," *Psychological Review*, 63(2), pp. 129-138.

Sobb, T., Turnbull, B., and Moustafa, N., 2023, “A Holistic Review of Cyber-Physical-Social Systems: New Directions and Opportunities,” *Sensors*, 23(17), 7391.

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), 5319.