UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

DISTRIBUTED PRODUCT DEVELOPMENT APPROACHES AND SYSTEM FOR ACHIEVING OPTIMAL DESIGN

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

By

ZHIQIANG CHEN
Norman, Oklahoma
2006

UMI Number: 3238558

UMI Microform 3238558

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

DISTRIBUTED PRODUCT DEVELOPMENT APPROACHES AND SYSTEM FOR ACHIEVING OPTIMAL DESIGN

A DISSERTATION APPROVED FOR THE
SCHOOL OF AEROSPACE AND MECHANICAL ENGINEERING

BY

____
Dr. Zahed Siddique, Chair

____
Dr. Kuang-Hua Chang

____
Dr. Kurt Gramoll

____
Dr. Shivakumar Raman

____
Dr. Alfred G. Striz

ACKNOWLEDGEMENTS

There are a number of individuals that have made contributions to this work. First of all, I would like to thank Dr. Zahed Siddique for his advice and patience, which makes this dissertation ever possible to be finished. This research has been continuously supported under a funding provided by US Tinker Air Force, in particular I would like to thank Dr. Kuang-Hua Chang for his support, advice and constructive suggestions.

I would like to express my appreciation to all faculty and staff members in School of Aerospace and Mechanical engineering at University of Oklahoma who provided a nourishing environment for my study and research.

More importantly, I would like to thank my committee: Dr. Zahed Siddique, Dr. Kuang-Hua Chang, Dr. Kurt Gramoll, Dr. Shivakumar Raman, and Dr. Alfred G. Striz. They took their valuable time to review my course requirements, research proposal, and this dissertation. Their comments have been very helpful in improving the quality of this dissertation.

Last, but not least, I want to thank my wife Nan, for enduring my preoccupation with this work and accompanying me through all the tough times.

CONTENTS

ABSTRACT

New paradigms, along with accompanying approaches and software systems are necessary to support collaborative design work, in a distributed design environment, of multidisciplinary engineering teams who have different knowledge, experience, and skills. Current research generally focuses on the development of online collaborative tools, and software frameworks that integrate and coordinate these tools. However, a gap exists between the needs of a distributed collaborative design paradigm and current collaborative design tools. On one side, design methodologies facilitating engineering teams' decision making is not well developed. In a distributed collaborative design paradigm, each team holds its own perspective towards the product realization problem, and each team seeks design decisions that can maximize the design performance in its own discipline. Design methodologies that coordinate the separate design decisions are essential to achieve successful collaboration. On the other side, design of products is becoming more complex. Organizing a complex design process is a major obstacle in the application of a distributed collaborative design paradigm in practice. Therefore, the principal research goal in this dissertation is to develop a collaborative multidisciplinary decision making methodology and design process modeling technique that bridges the gap between a collaborative design paradigm and current collaborative design systems.

In this dissertation, three major challenges are identified in realization of a collaborative design paradigm: (i) development of design method that supports multidisciplinary

design teams to collaboratively solve coupled design problems, (ii) development of process modeling techniques to support representation and improve complex collaborative design process, and (iii) implementation of a testbed system that demonstrates the feasibility of enhancing current design system to satisfy with the needs of organizing collaborative design process for collaborative decision making and associated design activities.

To overcome the first challenge, decision templates are constructed to exchange design information among interacting disciplines. Three game protocols from game theory are utilized to categorize the collaboration in decision makings. Design formulations are used to capture the design freedom among coupled design activities.

The second challenge is addressed by developing a collaborative design process modeling technique based on Petri-net. Petri-net is used to describe complex design processes and to construct different design process alternatives. These alternative Petri-net models are then analyzed to evaluate design process alternatives and to select the appropriate process.

The third challenge, implementation of collaborative design testbed, is addressed by integration of existing Petri-net modeling tools into the design system. The testbed incorporates optimization software, collaborative design tools, and management software for product and process design to support group design activities.

Two product realization examples are presented to demonstrate the applicability of the research and collaborative testbed. A simplified manipulator design example is used for explanation of collaborative decision making and design process organization. And a reverse engineering design example is introduced to verify the application of collaborative design paradigm with design support systems in practice.

The research in this dissertation attempts to provide theoretic approaches and design systems to support engineers who are located in different places and belong to different teams or companies to work collaboratively to perform product development.

CHAPTER 1
INTRODUCTION

Engineering design could be regarded as a transformation process from a set of functional specifications and requirements into a complete description of a physical product or system, which meets those specifications and requirements [1]. Design and development of a product requires considering different aspects of the product through coordination, negotiation, and discussion in a collaborative environment. A design engineer considers the product to function efficiently and reliably; a production engineer considers manufacturing the product in large numbers, quickly, cheaply, accurately and with the lowest possible number of defects; an entrepreneur invests in new products and expect an attractive return. Each participant plays a role as a stakeholder, generating information from his/her viewpoints or perspectives which influence the design through his/her design decisions. Collaboration is essential in a design process to avoid decision making mistakes, to shorten design time, and to improve design quality.

In addition to collaborative decision making between stakeholders, product design requires multiple participants to be involved to perform various collaborative design activities. For example, in the process of design concept generation, participants such as design managers, design team members, engineers, marketers, and even customers are asked to contribute their efforts in the design activities. The cooperation of multiple participants can greatly accelerate the design progress.

In recent years, more and more companies have their design resources distributed among different geographic locations. The studies of collaboration among distributed designers are required to achieve successful distributed design during product design and development. Specific focuses on design collaboration in this dissertation are: (i) Achieving collaborative decision making; (ii) Modeling collaborative design process; and (iii) Implementing design system to support real-time and synchronized group design activities.

1.1 Mechanical Design in a Distributed Environment

1.1.1 Understandings of Mechanical Design

There are many existing approaches focused on different aspects of engineering design process from various viewpoints. Researchers, engineers, product managers usually have different viewpoints of what design process is and they have different approaches to help them understand the characteristics of design process. These approaches can be generally classified into three groups [2]. The first group, mainly developed by engineers, focuses on investigation of how technical design decisions are made to establish systematic design methodologies. Design process models are often implied in the associated design methodologies and theories. The design theories provide the guidelines for designer to make technical decisions more consciously and systematically [3]. The second group comes from business operation and project management research. This group views design process as workflow with task dependencies and product information exchange. From this aspect, design is modeled as information driven processes among design activities. Design organization is viewed as a stochastic processing network in which engineering resources are "workstations" and design tasks are "jobs" that flow among them. The third group comes from CAD and CAE areas, which view collaborative design as individuals accessing product data and sharing the design information. Design process is accordingly specified as the management of the product data.

In this dissertation, our first focus is decision making which will be addressed in Chapter 4. The research goal is to enhance collaboration by achieving collaborative decision making, in which engineers solve the design problems in their own disciplines through proper interactions with other designers/participants. Compared with current research works of decision making for coupled design activities, the approach presented in this dissertation does not integrate the design activities in various engineering disciplines. It divides product development into separate design activities in various engineering disciplines and tries to achieve design collaboration by applying proper interactions for engineers from different disciplines. There are some existing research works following the philosophy of collaborative decision making and most of them can be categorized into the area of game theory based design approaches. The decision making approach in this dissertation improves current game theory based mechanical design approach by providing a mechanism to manage the design freedom to solve coupled design activities.

Besides classifying engineering design process into different aspects, mechanical design activities can be analyzed into different levels. Generally there are seven levels to describe a mechanical design activity.

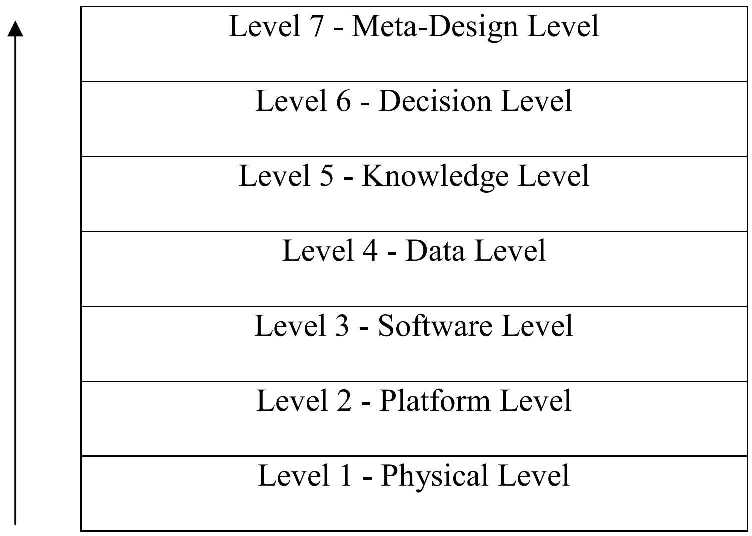


Figure 1.1: Levels in product development activity

The physical level contains basic hardware components in a product development environment, including computers, equipments and communication networks, etc. The platform level is the operating systems running on hardware. The software level consists of engineering CAD/CAE/CAM software, communication software, etc. Basic engineering operations are accomplished at this level. The data level consists of the input/output of the software. At this level, file format transform, database operations are accomplished, etc. The knowledge level represents information about product development, including the variables, goals, bounds, constraints, etc. within the activities. At this level, the engineering team collects and translates information into decision making knowledge such as mathematical formulations, computing equations, parameters, etc. The decision level contains decisions made by the engineering team. The meta-design level represents the framework of the entire product realization process. Collaboration and coordination of the engineering teams' decision making activities are accomplished at this level.

The focus of research or topic is different at each level of product realization activity. For instance, at the physical level, researchers are interested in the organization of equipment and material flows within a working environment; at the software level, researchers study how to develop robust and efficient software, or how to ensure the compatibility between software packages, etc.; while at the decision level, methodologies are developed to sustain decision making activities during product realization; and at the meta-design level, approaches that can organize design process efficiently and enhance collaboration and coordination between separate activities are developed.

In this dissertation, our second focus is meta-design level approach, which is used to guide high level design activities and will be addressed in Chapter 5. Engineers apply meta-design level approaches to organize a design process which consists of decision making in various engineering disciplines and explicitly describes the interactions of engineers' decision making. Compared with current research works, the design process modeling approach presented in this dissertation is the first Petri-net model which describes the relationship of design activity, design decision and design variables in a design process. The Petri-net model presented in this paper provides the possibility for engineers to explicitly describe the overall design process as well as detailed design information. In this dissertation, design variables equal to design parameters that can be adjusted for better performance. In some previous research, Petri-net is used to model general design tasks in a design process, which do not contain the detailed design information such as the exchange of the values of design variables.

Besides decision and meta-design level, software level design is also a focus in this dissertation. Software level design aims at providing engineers with various design support systems and tools. With the help of these systems and tools it is possible to share design information, schedule design process so that a group of distributed engineers can work together. Software level design is the prerequisite condition for applying any design approaches into the distributed mechanical design and in this dissertation after design approaches of decision making and design process organization are introduced basic design systems and new design tools are developed so that a typical distributed mechanical design based on the approaches introduced in this dissertation can be supported. All these three levels are tightly related with design collaboration in a distributed mechanical design. Software tools are required to accomplish many group design activities as well as individual activities. In this dissertation, design tools are developed for some specific design activities such as House of Quality creation. These real-time tools are important for applying distributed mechanical design but are not provided by current researchers and commercial companies.

The emergence of computer and network technology has provided opportunities for researchers to construct and build systems to support dynamic, real-time and seamless engineering design in a concurrent manner within a distributed environment. Numerous research efforts are mentioned in this field, which are resulting in continuous advancement and evolution of new approaches and tools. In this dissertation, the research motivation is to provide an answer of how to perform decision making, engineering design activities and design process organization in a distributed mechanical design based on research approaches and existing information technology? The potential to improve current research to achieve better distributed mechanical design is explored in this research.

1.1.2 Traditional Mechanical Design

A product realization approach is depicted in the form of the organization framework of the activities, called activity architecture, which represents engineering teams' philosophy for product realization. Traditionally the design and manufacturing process is organized in a sequential architecture which follows the trial and error approach and allows multiple design iterations to identify the proper solution from design alternatives. This sequential architecture has wide applications in industry, and many companies use it to find optimal design. The advantage of a sequential architecture is the ease of management of the process - downstream activities are not activated before upstream activities have been finished. Coordination between different engineering disciplines which include engineers having certain specialties are not always required, since usually there is a centralized management department existing in a traditional design process. The centralized management department monitors and controls the entire engineering process step by step. Although a sequential architecture manages design activities in a straightforward manner, it usually takes a relatively longer time to design a product and causes design iteration at later design stages when the cost of design changes are prohibitively high.

By overlapping upstream and downstream activities, parallel architecture is an improvement to the traditional one. It saves design time and product development cost due to the early identification and correction of errors. With the overlapping of downstream activities in early design stages, design changes that usually occurs at later design stages can be identified at early stages when engineers still have the flexibility to change the product design with low cost.

Concurrent Engineering (CE) is another widely used design approach. In CE, engineers from different disciplines consider the product design from various aspects. Product development team considers the information of downstream activities at the early stages. Design defects can be recognized in the early design stage when cost of design changes is still low and engineers have the design freedom to make these changes. As a result, CE reduces the likelihood of time consuming iterations that often happen in the sequential and parallel product design and development.

One common characteristic of all the above mentioned activity architectures for product development is that in all these architectures there is a centralized management team which controls the entire product development process. The centralized management team plays the role of a controller for coordination and communication among multiple engineering disciplines. Centralized management is relatively simple for organization because it assumes the centralized management team integrates all design information and therefore is able to make proper multidisciplinary design decisions that considers the results of late product development stages. In practice, the assumption of centralized management is usually impractical. Furthermore, centralized management team often forms a bottleneck of information exchanges among engineering disciplines. Any design modification need to be reported to the centralized management team and the centralized management team forces corresponding disciplinary design teams to make design changes in response. By applying distributed activity architecture in product development, we expect to remove the bottleneck of centralized management team and thus increase design efficiency.

1.1.3 Distributed Mechanical Design

Since the 1990's, the requirements for shorter time, lower cost, and higher quality lead the challenges in product development processes in which concurrency and distribution are important [4]. Although CE has been accepted as an approach to improve product development processes, the globalization of economy requires further enhancement of collaboration among distributed product development resources. Many product development companies have established their overseas branches and plan to implement their product development locally to achieve quick response to the change of product market.

In this dissertation, "distributed" has different meanings when different levels of mechanical design are referred to. In data level of design, "distributed" means data are not located in same site. In decision level of design, "distributed" means design decisions are made to solve single design activity not global integrated design activities. In Meta-design level, "distributed" means there is no centralized management team that controls whole product development.

With the fast development of information technology in the last twenty years, building a distributed product development environment have become possible for many companies. As an example, first design and manufacture data are converted from traditional paper to digital data. Compared with paper data, digital data is easy to transfer using computer network to remote sites. Second, Product Lifecycle Management (PLM) has been adopted by not only large companies, but also by some small companies who find PLM is helpful to manage numerous overlapping projects with small quantities of products. In an enterprise environment, PLM is used to manage product data, development process, and different design resources. It provides a base system for engineers in different geographic locations to work together. Combination of digital design and manufacture, PLM, and broad bandwidth computer network technologies provide the possibility for to achieve successful distributed product realization.

From research works, distributed design has the following characteristics that can be used to differentiate it from traditional product design:

1. Engineering teams are geographically dispersed.

2. Engineering software are heterogeneous and installed on different types of platforms.
3. Data exchange, engineering communication and coordination are available for most of distributed team members.
4. Centralized team does not exist or manage whole product development.
5. Product development process is a dynamic, flexible, and ever changing process that is adapted to the real time product development situation.

To support product development with the above five characteristics, engineers need the support from physical level of mechanical design such as computers and network devices, platform level of mechanical design such as network operating systems, software level of mechanical design such as various group design tools, data level of mechanical design such as product data management, decision level of mechanical design such as collaborative decision making approach and meta-design level of mechanical design such as collaborative design process modeling approach. The definition of distributed mechanical design is given as:

> **Distributed mechanical design** is a systematic approach governing product development. It is developed to support geographically dispersed multidisciplinary engineering teams to work collaboratively to solve product development problems and tasks. Each of these engineering team has discipline oriented tools, knowledge and different design goals, constraints and design parameters. A systematic approach is required for supporting different engineering teams to make collaborative design

decisions, organizing these teams to work in a collaborative process with proper coordination and communication.

1.2 Challenges of Distributed Mechanical Design

The basic introduction to the various understandings of mechanical design in Section 1.1 and the difference between traditional and distributed mechanical design indicate the important challenges to realize effective distributed mechanical design based on existing computer software and network technology. In a distributed design environment, achieving successful design has three challenges which are related with three aspects of design namely decision making, activity workflow, and product data management. The dissertation presents these challenges and provides approaches and design systems to help overcome these challenges to realize decision making, design activity organization and product data management in an environment where engineers are placed in different locations.

Corresponding to the challenges of distributed mechanical design, most of the current research can be categorized into three areas. In software level, online engineering tools and Group Design System (GDS) are developed to support product data management and implementation of various group design activities. In decision level, systematic approaches are required to implement multidisciplinary decision making. In meta-design level, design process modeling approach is necessary to organize decision making and other design activities to form a design process.

1.2.1 Challenges in Multidisciplinary Decision Making

Traditional multidisciplinary design is focused on the aspect of design optimization, the idea of formulating a design problem in rigorous mathematical terms, and mathematically tracing a path in the design space from the initial toward improved designs. During the past two decades much progress has been made in numerical optimization that offers the possibility for researchers to solve relative complex design problems. Current optimization techniques can handle tens of thousand, or even hundreds of thousands of variables. Optimization variables for Nonlinear Mathematical Programming algorithms can also go beyond a few hundred to describe a design in some cases. However, as number of design variables keeps on increasing in some product developments, formulating a design problem, making an integration of engineering considerations and solving it become a more and more difficult task in multidisciplinary mechanical design. The interdisciplinary interaction (coupling) tends to present additional challenges beyond those encountered in a single discipline problem [5].

The most widely practiced approach to handle these challenges is by integrating all the decision making through a system level engineering team [6]. This system level design team controls the information communication, handles interdisciplinary interactions, and makes the design satisfying with product requirements. CE is a philosophical idea guiding this kind of system level integration and synthesis, which becomes difficult in some complex multidisciplinary problems such as aircraft design in which thousands of design variables are reported in only one discipline [7]. There are many research [8] that attempt to remedy this complexity burden, which is exponentially increased as various engineering disciplines are involved during product development. Some related research areas are Multidisciplinary Optimization (MDO) and collaborative decision making.

As the researchers keep on pursuing the design of high performance products, balancing product performance considerations with manufacturing, economics, and life cycle issues, two obstacles have been met for a multidisciplinary optimization. One is computational burden and the other is the organizational challenges [9]. Numerical optimization capabilities lag in comparative fidelity as characterized by the number of variables describing a design for optimization and for analysis (simulation) [10]. The computation cost of large-scale computationally expensive models for high fidelity analysis in disciplines is an obstacle for many engineering practices, not to mention the iterations in analysis of coupled systems. Another obstacle for applying MDO is the organization challenge. Forming multidisciplinary design optimization problems needs the cooperation of engineering teams in multiple disciplines. The implementation of MDO is sometimes restricted by analysis code, data, and human team organization; incompatibility between disciplinary analysis codes; complexities in software integration; and defining the roles to be performed by the various departmental design teams.

To overcome the two obstacles of MDO, one research direction is to develop more flexible MDO architectures that tackle problems with broad coupling. The progress in this research direction has lead to the Concurrent SubSpace Optimization CSSO approach

[11-13] and Collaborative Optimization (CO) approach [14-16]. In CSSO, introduced by Sobieszczanski-Sobieski, the discipline level teams solve the sub-problems concurrently while system level team coordinates the conflicts and achieves multidiscipline feasibility. In the solution of sub-problems, the non-local constraints are approximated using Global Sensitive Equation, and "responsibility" coefficients are assigned as constant parameters in sub-problems to reflect the local influences of non-local constraints. System level team updates the responsibility coefficient after each round of local decision making until convergence [11]. In CO, introduced by Kroo and Braun, auxiliary variables are introduced to replace the coupled variables in each sub-problem so that they can be solved concurrently. The objective in each sub-problem is minimizing the "discrepancy" between the auxiliary variables and coupled variables, usually formulated as a least square function, while system level problem is formulated to satisfy the system requirements and minimize the overall discrepancy, termed as "interdisciplinary compatibility constraints" [17-18].

Although these approaches decompose a MDO problem into separate disciplinary optimizations and address the needs of multiple disciplines, there are not many industrial applications to prove that the two approaches can resolve the issues of computational cost and organization challenges in all complete industrial scale product designs. It has been recognized that a total cooperation among disciplines in a CE environment is rare in practice [19]. CSSO, CO and other bi-level approaches still suffer from the so-called "curse of dimensionality". That is when number of coupled variables increase, the system level problem will become difficult to be solved.

Recently some researchers have focused their works on the human aspect of the multidisciplinary design. Instead of integrating engineering disciplines and managing all design decisions in a centralized team, supporting distributed team members to make individual design decisions with proper coordination and communication can also help find optimal solutions for coupled design problems. From this viewpoint, product design is a process of going forward by continual question-answer iterations. In order to answer these engineering questions iteratively, the key issue is how engineers can make their decisions separately without full cooperation with other engineers in different disciplines? The engineering process can be viewed as a series of relative independent decisions which gradually define a new product in more and more detail. The research object is to develop a mathematic construct that can model the degree of design freedom in collaboration so that independent decisions can be made. The philosophy of this research differs from the aforementioned multidisciplinary design. With the awareness of the challenge to achieve full cooperation in design, an alternative way is to separate the disciplinary design with certain degree of freedom. A significant progress in collaborative decision making area is the introduction of game theory [20-21] which forms the foundation for this research. In Lewis and Mistree's research, multidisciplinary design is abstracted as a set of games. Engineers in each discipline are the players. Based on the analysis, simulation, or other obtainable information the players play the games or in other words make their design decisions to maximize their own game rewards [22]. In game theory, there are three game protocols, cooperative, noncooperative and leader/follower. Each protocol models a game construct that represents one type of interaction among engineering teams. In this dissertation these protocols are termed as design collaboration strategies. Based on the game based design approaches, the method of collaborative decision making in this dissertation is developed. The research questions include how to represent an engineering problem with coupled design variables, how to solve the problem separately and how to maintain the design freedom so that it is possible for other designers to find solutions in their design problems.

1.2.2 Challenges in Design Process Modeling

Besides the multidisciplinary decision making approaches, it is important to organize the overall design process which consists of various engineering disciplines and their interactions. In current research, the major difficulty for organizing a design process is complexity. Especially in CE practice [8], when product development involves numerous engineering disciplines and the overall process becomes difficult to be described, analyzed, and improved. The complexity is the reason that most of the existing design process modeling approaches are text based, where a text description of the characteristics of a design process in each stage [23-24] is provided or brief information is provided for engineers to understand the task dependency relationship in the design process [25-28].

In the research, a new idea of modeling a distributed mechanical design process is provided. Three aforementioned game protocols in section 1.2.1 are treated as three design collaboration strategies and the design process is formed by selecting different combinations of three design collaboration strategies to organize all coupled design activities. Our approach is developed to help engineers understand the overall design process and the exchange of detailed design information. To overcome the organization challenge in multidisciplinary design, Petri-net, a graph modeling technique, is applied in this dissertation to model the complex dependency relationship and capture the key information of dependent relationships in a design process.

1.2.3 Challenges in Online Engineering Tools and Group Design Systems

Although in theoretic studies researchers have extensively discussed the multidisciplinary decision making and design process modeling, in design tool development they showed more efforts on the detailed design activities, mainly implementing various Group Design System (GDS) or online engineering tools to support specific distributed mechanical design activities. Many researchers aim to provide these powerful design tools or systems to help distributed engineers work efficiently.

In a distributed mechanical design, it is essential to have various design activities (i.e., geometric modeling, engineering analysis, design information preparation, etc.) supported by a group design software tool. The implementation itself becomes a challenge to researchers because traditional design tools are developed for single user to work individually. Real-time group design tools supporting multiple users are developed using different software development tools and in development of these design tools additional issues such as data synchronization need to be considered. As to the functions of design tools, currently design software can only provide limited functions of geometric modeling and engineering analysis. Some new design tools developed need to support other design activities for example creating House of Quality, real-time geometric model based discussion tool.

As a part of this research, several design tools are implemented to support specific group design activities such as creating House of Quality, selecting design alternatives, and discussing design solution based on geometric model and a design system to manage product data and design process. The implementation makes it possible to build a testbed which has basic capacity to support distributed mechanical design and can be used to verify our ideas of collaborative decision making and design process modeling using practical scenarios.

1.3 Research Questions and Objectives

The above introduction highlights the research challenges that are fundamental for distributed mechanical design. Consequently, the research work in this dissertation starts from answering the questions:

(i) What are required to accomplish a distributed mechanical design?

(ii) How can distributed engineering teams make design decisions separately and achieve proper collaboration?

(iii) How can engineers organize a distributed design process?

(iv) What group design system is needed to facilitate engineers' design activities in a distributed design process?

Objectives of this research include: (i) develop a framework of distributed mechanical design (Question 1); (ii) develop a systematic approach that can be used to achieve collaborative design decision making (Question 2); (iii) develop a systematic approach that can be used to organize a distributed design process in the aspect of design decisions (Question 3); (iv) implement a test-bed that integrates design-oriented GDS and is capable of supporting and managing design activities including design decisions and some other design operations such as House of Quality creation, real-time model discussion, product data access and so on (Question 4).

1.4 Definition of the Needs

The needs of a distributed mechanical design stem from industry.

Needs from engineers:

Mechanical design engineers are faced with the tedious and time-consuming task of painstakingly running multiple simulations in an attempt to iteratively search an often elusive acceptable solution that satisfies most of the requirements. In CE, usually engineers focus on their specific work and have difficulty finding and understanding the design considerations of other engineers from different disciplines or teams. Designers make their decisions without adequate coordination. Very often engineers develop the design based on only a minimal set of the most critical design factors and neglect the rest, hoping any conflict could be corrected later in the cycle [29]. Consequently, costly rework is usually required with some portion of design work wasted. There is a need to develop systematic approaches and enhanced design systems to help engineers make proper decisions and be aware of any relevant design information so that design reworks can be reduced and conflicts can be avoided.

Needs from companies:

Some product failures can be attributed to radical shifts in the market or because companies are out of touch with customer requirements. A large portion of the failure rate is a result of designers being absorbed in engineering information related with their disciplines and not able to adequately address the shift of the market or engineering requirements that are critical to satisfy pressing market demands for today's complex products. There is a need to provide a collaborative design system so that design communication and data sharing are supported; the information about market and customers is properly delivered to engineers through product data sharing and design communications.

Needs from improving traditional design tools:

Computer network makes it possible to link the design resources in a distributed environment. However limitations of conventional computer tools, which are intended for single user, greatly lessen the design work efficiency. Cooperation with team members to perform design activities is not considered during the software development process. There is a need to provide group design tools to make it possible for engineers in different locations to perform group design activities together.

Needs from managing complexity of design process

One common problem in concurrent engineering is the "90% syndrome" [28], which describes a project that reaches about 90% completion on schedule but then stalls, finally finishing after about twice the originally projected duration. In the paper "Overcoming the 90% Syndrome: Iteration Management in Concurrent Development Projects" the authors suggested the approach of reducing the dependency of design tasks [28]. There is a need to reduce unnecessary design iterations. In this dissertation, design iterations are reduced by making collaborative design decisions using game theory protocols.

1.5 Overview of the Dissertation Organization

To facilitate the discussion of the dissertation, Figure 1.2 illustrates the organization of this dissertation. In Chapter 1, the research objectives were presented along with introduction of research challenges and motivations.

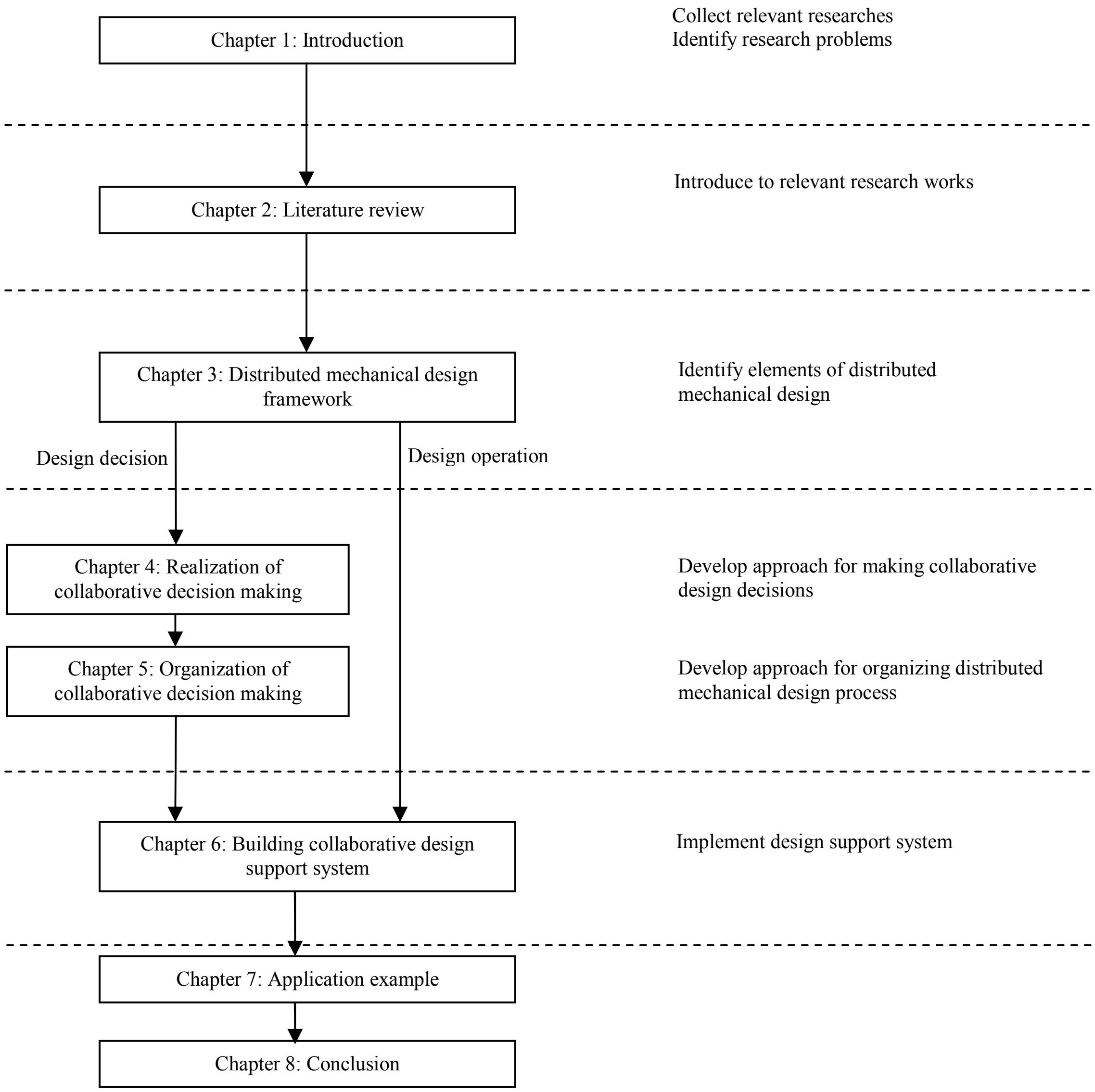


Figure 1.2: Organization of this dissertation

In Chapter 2, the related techniques or approaches are introduced. This chapter is a research background of this dissertation and introduces the useful research works that are conducted by other researchers and are used in this dissertation.

In Chapter 3, an overview of an ideal distributed mechanical design is given. Major elements of mechanical design are introduced. Approaches and system that are required to support these mechanical design elements are mentioned. From the discussion in Chapter 2, the constitutional elements of general distributed design are clarified. The rest of chapters in this dissertation are corresponding to these constitutional elements of distributed design and intend to make the elements available for engineers.

In Chapter 4, decision making is discussed. A collaborative design decision approach is presented to provide a solution for engineers to work individually and solve coupled engineering activities. Implementation of three game protocols in distributed mechanical design is illustrated. Formulation of maintaining design freedom is given.

In Chapter 5, another constitutional element, design process modeling is discussed. A design process modeling approach is presented which supports the meta-design activity. The developed approach is based on Petri-net and models the relationship of design decision activities in a distributed product development.

In Chapter 6, the purpose of research is to provide a design system with various design tools integrated. With all the approaches introduced in Chapter 4 and 5, distributed engineers still need a design system and design tools to help them accomplish their design tasks. The requirements of the design system and tools are presented based on the depiction of distributed mechanical design in Chapter 3. The developing techniques, system architecture and components are presented in the chapter to provide a more detailed introduction on how the design system works.

Chapter 7 is an example of applying the developed approaches, design system and tools. In this chapter, two design examples are presented. The examples show the detailed steps of collaborative decision making, design process modeling and design scenario based on the design system and tools. Chapter 8 is a conclusion of the research in this dissertation and highlights some of research contributions.

CHAPTER 2
LITERATURE REVIEW AND BACKGROUND

In this chapter, more details about the foundations of this research are introduced, along with literature review of related research. As mentioned in Section 1.1.1, collaborative decision making and design process modeling are two of three research focuses for achieving effective collaborative mechanical design. Collaborative decision making is the area of studying related to how two or more engineers can find solutions for coupled design activities. In this dissertation, the background research elements related with multidisciplinary decision making are Compromise Decision Support Problem (c-DSP) and game theory. In the Section 2.1, c-DSP is introduced which facilitates designers to formulate engineering activities. c-DSP is used in this dissertation to model engineering teams' design activity. The solution of c-DSP is recognized as the decision of the team. In Section 2.2, game theory is introduced, which facilitates to categorize the design collaboration strategies. Game theory provides an understanding of distributed mechanical design. Based on the game theory, distributed mechanical design is considered as a special game which requires the involvement of multiple teams from different disciplines.

Design process modeling is the area of studying the overall mechanical design process characteristics, which includes various engineering considerations in each design stage, design tasks, dependency relationship of design tasks, etc.. In this dissertation, design decision making in a distributed product development environment is modeled using Petri-net. Background information on Petri-net is presented in Section 2.3. Petri-net is originally designed to model concurrent systems, in this dissertation, it is has been extended to model the dependency relationship of collaborative design decisions.

2.1 Decision Support Problem Technique

"Designing is a process of converting information that characterized the needs and requirements for a product into knowledge about the product" [32-34]. In this definition, the product represents not only an artifact, but also the product realization process in more general sense. From the philosophy of Decision Based Design (DBD) product realization process is recognized as a set of design decisions which define the uniqueness of the developed product [32-40]. DBD is proposed to emphasize a different perspective of product realization from product to design process. As a rigorous approach to engineering design, DBD has recognized that decisions play a substantial role in engineering design and are largely characterized by uncertainty and risk [41].

In DBD, the principal role of a designer is to make design decisions. It is design decisions that convert the design concepts to design solutions. Decision making is an important aspect of mechanical design and provides a starting point for developing design approaches. These design approaches are generated based on design decisions made by designers or engineering teams. DBD is different from the computer based design approaches that are assisted by computer-aided design software, as well as optimization software or specific analysis tools such as finite element analysis tools. Even though decisions can be made based on many things including the results of engineering software, it is the decisions themselves that mark the progression of a design from initiation to implementation to termination. Decisions help bridge the gap between an idea and reality. They are a unit of communications that are characterized by information from many sources and disciplines and may have both discipline-dependent and discipline independent features. In DBD, it is the making of decision that causes the transformation of information into knowledge. The characteristics of decisions, which greatly affect the tone of our research, are governed by characteristics associated with the design of real-life engineering systems [42].

1. Decisions in design are invariably multileveled and multidimensional in nature.
2. Decisions involve information that comes from different sources and disciplines.
3. Decisions are governed by multiple measures of merit and performance.
4. All the information required to make a decision may not be available.
5. Some of the information used in making a decision may be hard, analysis-based, and some information may be soft, insight-based.
6. The problem for which a decision is being made is invariably loosely defined and open. Virtually none of the decisions are characterized by a singular, unique solution. The decision solutions are less than optimal and are called satisficing solutions.

The implementation of DBD can take many forms. One of the implementation approaches is Decision Support Problem (DSP) technique which offers support for human judgment in designing systems that can be manufactured, maintained, recycled, etc. DSP technique consists of three principal components: a design philosophy, an approach for identifying and formulating DSPs [43] and the software for solving the DSPs [44]. The DSP technique requires that product design be implemented in two phases:

1. Meta-Design. In this phase, the design process itself is designed wherein the product realization problem is partitioned into its elemental DSPs and a plan of action is devised, using discipline independent approaches. This phase represents the meta-design level in product realization activities.
2. Design. In this phase, the design process is implemented and the DSPs identified in former phase are formulated, solved and validated. This phase corresponds to the decision level in product realization activities.

DSP provide a means of modeling decisions encountered in design and the discipline specific mathematical models implementable on a computer are called templates. Multiple objectives, quantified using analysis based and experience based information, can be modeled in the DSPs. In the early stages of product realization, DSP technique can help the engineers questing for a superior solution of a design problem even when analysis based information is not available. While in the computer assisted environment this support is provided in the form of optimal solutions for DSPs. DSP has been used in variety of domains, including design of complex product like ship [36] and aircraft [45], design for manufacture [46], mechanical system, etc. Formulation and solution of DSPs provides a means for making the three types of decisions, selection, compromise and hierarchical. Among these three types, Compromise decision is used to improve a feasible alternative through modification. Because of the capability of handling tradeoffs of multiple goals, the compromise multi-objective decision model is suitable to represent multidisciplinary design problems [33]. In this dissertation, we select the compromise DSP to represent trade-off decisions in design problems.

C-DSP is a multi-objective decision model which is a hybrid formulation based on mathematical programming and goal programming [37, 44] to satisfy a set of constraints

Given
An alternative to be improved, domain dependent assumptions
The system parameters: n number of system variables, q inequality constraints, p + q number of system constraints, m number of system goals,
$g_k(\mathbf{X})$ system constraint functions ,
$f_j(d_j)$ function of deviation variables to be minimized
Find
System Design Variables**,** X_i i = 1, …, n
Deviation Variables, d^-_i, d^+_i j = 1, …, m
Satisfy
System constraints (linear, nonlinear)
$g_k(\mathbf{X}) = 0$; k = 1, .., p
$g_k(\mathbf{X}) \geq 0$; k = p+1, .., p+q
System goals (linear, nonlinear)
$A_j(\mathbf{X}) + d^-_j - d^+_j = G_j$; j = 1, …, m
Bounds
$X_i^{min} \leq X_i \leq X_i^{max}$; i = 1, …, n
$d^-_j, d^+_j \geq 0$, $d^-_j d^+_j = 0$; j = 1, …, m
Minimize: deviation function
$\mathbf{f} = [\ f_1(d^-_1, d^+_1), ..., f_m(d^-_m, d^+_m)\ ]$ (Pre-emptive)
$\mathbf{f} = \sum W_j(d^-_j + d^+_j)$ where $\sum W_j = 1$, $W_j > 0$ (Archimedean)

Figure 2.1: C-DSP Formulation

while achieving a set of conflicting goals as well as possible. The mathematical form of the c-DSP is given in Figure 2.1, the system and deviation variables, constraints, goals, bounds and deviation function are described in [29, 44]. In this dissertation, design variables equal to parameters that can be adjusted to improve product performances. The formulation shows that there are a set of goal G_i. The object function attempts to minimize all goal deviations to achieve a compromise design solution.

Currently, two objective functions are mostly used in formulating a C-DSP, the Archimedean solution scheme and preemptive approach [47] at evaluates a solution on the basis of preference.

A solution to a c-DSP is called a satisficing solution. "Satisficing" is a term coined in the context of optimization, meaning not the best but good enough [48]. The solution of the c-DSP is a point selected within feasible design space based on its degree of satisfaction to a set of conflicting design goals. Satisfaction is evaluated using the value of the deviation function in the c-DSP. The engineering team that makes decision has to tradeoff between the desired goals. It depends on the team to decide whether accepting the solution of a c-DSP or further investigating the problem by modifying the desired goals or feasible design space.

2.1.1 Why DSP Technique

The DSP technique provides a foundation for our research of decision making approach. The DSP technique plays a role at supporting the team to make appropriate decisions. It provides a clear representation of engineering design activities and its mathematical form can be solved using optimization software. In a distributed mechanical design, DSP is very helpful to formulate engineering design activity with cross-disciplinary design considerations. The DSP representation of design activity is the first step for design information exchange, problem solving and design process modeling.

The c-DSP is used to perform tradeoff studies of multiple design goals, which are typical in product realization activities. Team's design objective and requirements are modeled as design goals, and a deviation variable shows whether a specific target value G_i of a goal is met; and the difference between the target value and achievement A_i; the team's tradeoff strategy is clearly shown in the formulation of deviation function which is a function of d_i^- and d_i^+. Furthermore, collaborative product realization requires tradeoff between the teams. Each team controls a certain set of system variables and has different priority to make decisions. Therefore, the tradeoff strategies employed by engineering teams can be efficaciously modeled using c-DSP. The c-DSP has been tested and proved to be efficacious at representing the engineering teams' decision related design activities [43-45].

A c-DSP is capable of representing the decision making information in a design process. A team's decision consists of a design space, design objective and a tradeoff strategy of these objectives. The c-DSP provides a standard and disciplinary independent format that can be used to represent the decision making information in a product development activity. Moreover the dependence relationships between activities are represented with a set of system variables which are shared by several c-DSPs, called coupled variables.

Therefore, a set of c-DSPs can be used to represent the decision making information within a design process [48].

Consequently, in this dissertation, the mathematical formulation of c-DSP is used to convey decision making information between multidisciplinary engineering teams. Decision making information is the key for engineering communication. In distributed mechanical design, a standard and understandable information media is needed to represent the decision making information of each design activity. Following this idea, complex information exchange in a design process can be simplified into conveying an information package between teams. c-DSP is used as an information media, named as design activity template in the research of this dissertation. Due to its standard format and its capability of representing the decision making information in a design activity, a design process can be decomposed [49] and modeled into a set of c-DSPs. After modeling the product design activities, the next step is to solve these DSPs while keeping the activities separated. Game theory protocols are used to facilitate collaboration of decision making between the separated engineering teams [50].

2.2 Game Theory Protocols

The strategies of collaborative decision making in this dissertation are introduced from the game theory. Rao successfully applies cooperative protocol in multi-objective structure optimization [51]. Petriaux and colleagues combined game theory with genetic algorithm reduce the computing time in complex optimization problems [52]. Badhrinath and Rao present multiple player game and use leader/follower protocol to represent the interaction relationship between product design and manufacturing [53]. Lewis and Mistree illustrate the use of the principles of game theory to model the interaction between engineering teams in decision making and systematically study all three protocols [54].

As an example, imagine two designers u_1 and u_2 from two disciplines each controls the design variables x and y, and are minimizing their respective deviation functions $f_1=x^2+y^2$ and $f_2=|1-x-y|$. It is assumed that each designer represents his/her design problem using c-DSP formulation. According to game theory, a game consists of multiple players, strategy space for each player, and payoff function for each strategy [55]. Game theory can be used to model the above two player scenario, where designers u_1 and u_2 from two disciplines are treated as two players; design variable x and y are treated as strategy spaces; and deviation functions f_1 and f_2 are treated as payoff functions. When the collaborative design is treated as a game, there are three strategies that can be used to model various collaborative design scenarios [54], these strategies are: Pareto cooperative, Nash noncooperative and Stackelberg leader/follower.

Pareto cooperative strategy is a full cooperation model [55]. The assumption associated with this strategy is that each designer in one discipline has complete information and knowledge of the other disciplines so that they can achieve total cooperation with the other designers. Perfect communication and data exchange are provided to assure the availability of proper information. With Pareto cooperative strategy, coupling design problems are integrated, and the final cooperative design decision generates a Pareto solution because if the assumption of cooperative strategy is valid, designers are expected to obtain better design solutions when certain solutions exist that can improve the product design in all disciplines. Mathematically the result of the cooperative decision making is a Pareto solution (see Figure 2.2).

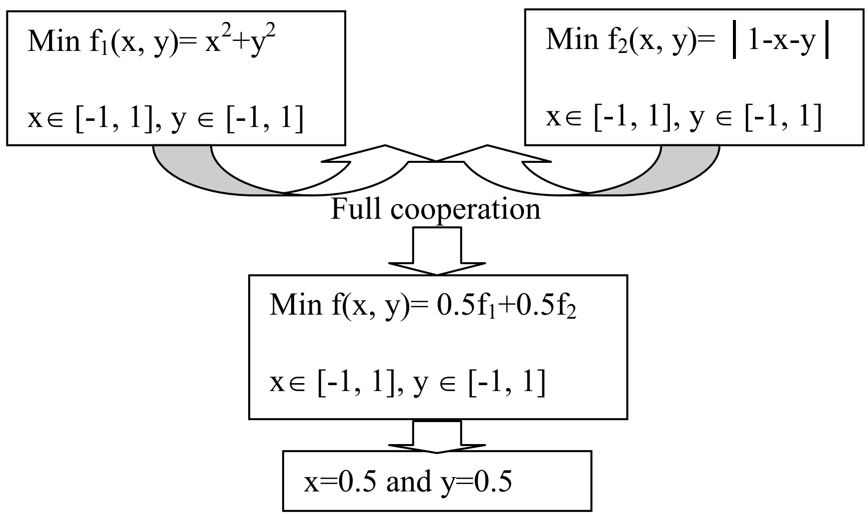


Figure 2.2: Pareto cooperative game construct

Nash noncooperative strategy occurs when design teams may not have proper information to make cooperative decisions. Designers make isolated design decisions with the assumption that other decision makers may adversely change the design to satisfy their own design objectives. The final noncooperative design decision reaches a Nash solution which is an intersection of all disciplines' Best Reply Correspondence (BRC) also called Rational Reaction Set (RRS) which is introduced in leader/follower protocol (see Figure 2.3).

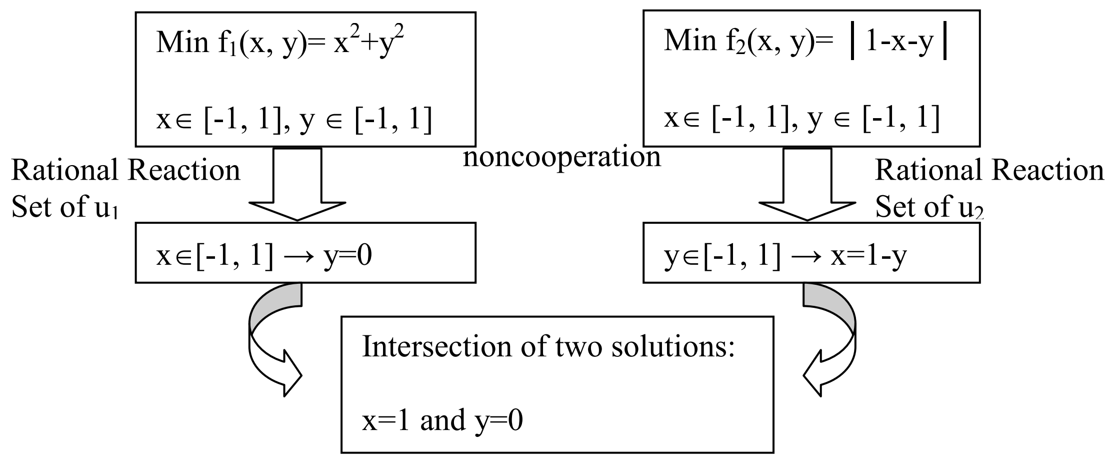


Figure 2.3: Nash noncooperative game construct

With Stackelberg leader/follower strategy, the leader makes its decision, finalizes the design, and then passes the information to the follower. The follower gets the information from the leader, attempts to solve the problem within leader's solution rules. Stackelberg leader/follower strategy works well when one designer dominates the decision making process or the "influence of a certain domain on another is strongly unidirectional" [54]. Full cooperation strategy is an ideal situation that rarely happens in practice because it is difficult to integrate a large-scale product design, especially in CE. Rather, in many product design cases, the Stackelberg leader/follower strategy is used more often. One difference of Stackelberg leader/follower strategy and sequential design is that the former uses a concept called the Rational Reaction Set [56]. Lewis [56] first introduced the Rational Reaction Set into collaborative design. In his work, Design of Experiment (DOE) technique is used to sample design solution points from the follower. These points are used to create response surface which is then fed to leader's design problem for seeking optimal design solution. The Stackelberg game of two players is illustrated in Figure 2.4, where Rational Reaction Set (RRS) of player u_2 is constructed as y_{21}=RRS(y_{12}). Through Rational Reaction Set leader's design space y_{12} works as the input to solve follower's response y_{21}. The leader u_1 uses the Rational Reaction Set y_{21} from the follower u_2 as a prediction of u_2's behavior. Based on the prediction, the leader u_1 makes design decisions that optimize the overall design without follower's cooperation. Although in this game construct, the follower first generates RRS, it is the leader makes a design decision first that specifies the acceptable solution ranges of design variables. In the example in Figure 2.4, the solution of designer u_1 without cooperation from u_2 is x=0, y=0. The solution of designer u_1 considering follower u_2's discipline performance is x=0.5, y=0.5. If the allowable solution range of leader u_1 is set as $x \in [0, 0.5]$, $y \in [0, 0.5]$, the final solution is x=0.5 and y=0.5.

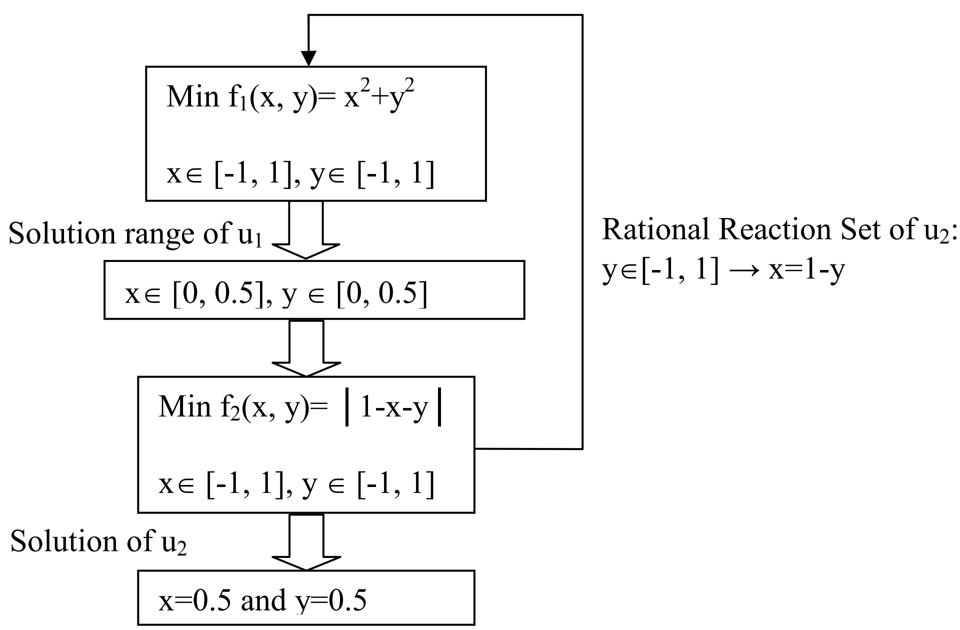


Figure 2.4: Stackelberg leader/follower game construct

Following the philosophy of game based design, researchers construct the design activities into Pareto, Nash and Stackelberg games, especially Stackelberg leader/follower game which is often used in mechanical design. c-DSP is a common technique to formulate the design activities for problem solving using Stackelberg leader/follower strategy. Chen [57] proposed a robust design approach in multidisciplinary design which instead of looking for a single point solution the leader looks for a range of solutions that provides design flexibility for the follower. Taguchi's robust design formulation has been applied in Chen's research about collaborative decision making [58-60]. By introducing the robust design formulation, in Chen' approach a leader is seeking a range of solution which is not only optimal for product performances but also robust to the design variations. c-DSP is used to find a solution that can satisfy both robustness and optimal performance requirements. In another research, [61-64] presented a design index approach which provides the follower with an evaluation of his/her decision. This evaluation is based on the mean and standard deviation calculated by the leader. One common problem for both research is that within the available solution range either robust solution range of Chen or index qualified solution range of Xiao, the follower tends to pick up the solution point that has the best performance in his discipline. This is not a tradeoff decision and what is more a leader cannot control the loss of his discipline performance once the design work has been passed onto the follower. Only option a leader can do is to shrink the available solution range and when he does this the follower's design freedom is reduced. The resolving of the above conflict is a research aim that this dissertation attempt to address.

2.2.1 Why Game Theory Protocols

"The modeling strategic and optimal behavior based on the actions of other individuals is known as a game and the study of the strategic behavior is game theory [20]. Marston investigated the influences of teams' behaviors on design result and presented the term game based design [65], meaning "a set of mathematically complete principles of rational behavior for designers in any scenario", which is composed of four essential elements: game theory, decision theory, utility theory and probability theory. In this dissertation, it is not expected that design likes a game in every aspect. However, some behaviors of the engineering teams can be modeled using game theory. In this dissertation, the research interest is "design using game theory". It is not necessary to require teams to behavior following every principle in game theory because some principles are not appropriate in the engineering world.

Game theory protocols are used to facilitate solving of coupled design activities. From the perspective of game theory, the design cooperation of multiple engineering teams can be treated as a game. The design collaboration strategies in a multidisciplinary mechanical design can be categorized as three basic game protocols. Based on different design collaboration strategies, different mechanical design can be formed. This forming process is called a game construct. Game construct can clearly discover the collaborative relationships of engineering teams and provides a possibility to model design activities that are coupled with complex dependency.

In this dissertation, three game theory protocols are used in the research of collaborative decision making in Chapter 4. Correspondingly, collaborative decision making can be classified into three basic types. In the research of design process modeling in Chapter 5, game construct based on three game theory protocols is used to reveal the relationship of design activities in a design process.

2.3 Petri-net System Modeling Technique

Petri-net is a graphical and mathematical modeling tool, which has been applied to model various systems. Petri-net has been a promising tool for describing and studying information processing systems that are concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic [66]. A Petri-net can be used to analyze state machine, communication protocol, parallel activities, data flow computation, synchronization controls, multi-processor systems, etc. In this dissertation, Petri-net is applied to represent the design decision making process.

Historically speaking, the concept of the Petri-net has its origin in Carl Adam Petri's dissertation [67], submitted in 1962 to the faculty of Mathematics and Physics at the Technical University of Darmstadt, West Germany. The dissertation was prepared while C. A. Petri worked as a scientist at the University of Bonn. Petri'swork [67, 68] came to the attention of A. W. Holt. The early developments and applications of Petri-nets (or their predecessor) are found in the reports [69, 74] associated with this project, and in the

Record [75] of the 1970 Project MAC Conference on Concurrent Systems and Parallel Computation. From 1970 to 1975, the Computation Structure Group at MIT was most active in conducting Petri-net related research, and produced many reports and theses on Petri nets. Most of the Petri-net related papers written in English before 1980 are listed in the annotated bibliography of the first book [76] on Petri nets. More recent papers up until 1984 and those works done in Germany and other European countries are annotated in the appendix of another book [77]. Three tutorial articles [78-80] provide a complemental, easy-to-read introduction to Petri-nets.

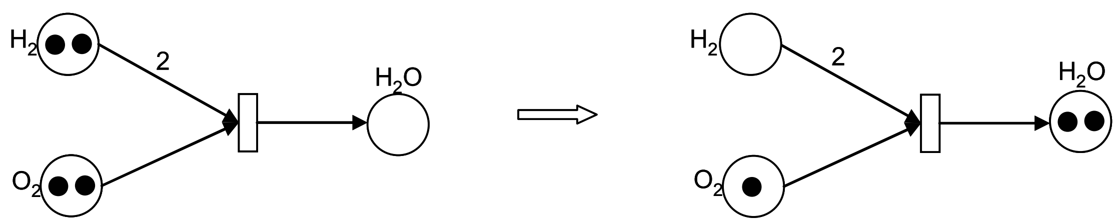


Figure 2.5: Petri-net Example (adapted from [10])

A Petri-net graph represents a process with two types of nodes: places and transitions. Using Figure 2.5 as an example, the three places are H_2, O_2 and H_2O. The transition is the chemical reaction that converts hydrogen and oxygen into water. Directed arcs join the places with transitions. Each place may contain one or several tokens represented by dots. In this example, token represents molecules. Later in Chapter 5, token is defined to represent design solutions. The following transitions of one place can only be executed when the required tokens are available. A weight can be associated with each connection of place and transition to represent the required tokens, which is a positive number. In example shown in Figure 2.5, two hydrogen and one oxygen molecules are required to fire the chemical reaction. The marking of the Petri-net is a vector that contains the number of tokens in all places.

Petri-net was first adapted into the field of collaborative design process modeling at the end of the 1990's. Since then several Petri-net design process models have been developed [81-83], however, these models attempt to provide analytical mechanisms for organizing design tasks with dependency relationship [82] or coordinating multi-user online design activities to avoid conflict [83]. These Petri-net models do not address the idea of modeling a design process in the aspect of design problem and decision making.

2.3.1 Why Petri-net System Modeling Technique

Petri-net has wide applications in the area of computer concurrent system modeling such as modeling computer network communication protocols. In the engineering field, its applications are varied in different research works. Similar with the research area of Group Design System, in the research of computer science department, there is a research area of Computer Supported Cooperative (CSCW). An application of Petri-net in CSCW area is to model multiple users' real-time software operations [84]. In some other research, Petri-net is used to model common design activity. The starting condition of design activity, the time of performing design activity and activity results can be modeled in a Petri-net model [85]. Petri-net provides a description of design process in the aspect design activity and can be used in design task scheduling.

In this dissertation, Petri-net is used to model a design process in the aspect of a special design activity, design decision. The model of Petri-net can clearly represent the relationships of cooperative design activity and decision making. the research works discuss the method of how to generate variety of Petri-net model for product development and how to evaluate different Petri-net design process models based on their engineering performances.

2.4 Design Process Modeling

The research on design process modeling presented next is categorized according to three groups. The first group of approaches comes from the engineering discipline, focusing on the investigations of how the design decisions are made. The second group, which is mainly from business operation, project management and CE process integration, considers design process as workflow with task dependency and information exchange. The third group is from computer science; they view the design process as product data storage, exchange, and management.

2.4.1 Decision Making Based Approaches and Theories

Design process models are often found in the research of design theories and methodologies, such as Systematic Design Model [86], Axiomatic Design Model [87], Quality Function Deployment [88], General Design Theory [89], etc. Systematic Design Model divides a design process into a sequence of design stages and discusses the decision making approaches for each stage. Axiomatic Design Model applies two fundamental principles that govern good design practice, which are maintaining the independence of the functional requirements and minimizing the information content of the design. Quality Function Deployment translates customer requirements to engineering specifications and becomes a link between customers and design engineers. These theories provide the guidelines for designers to make technical decisions more consciously and systematically [2]. The key issues of these theories are the rationales under technical decisions. However, the influences of designers' cooperation and collaboration on the final design results are not explicitly addressed in these traditional design process models.

2.4.2 Workflow with Task Dependency and Information Exchange

Researchers of this group view design as information generation and a conversion process produced by a set of design tasks or further design activities. Design organization is viewed as a stochastic processing network in which engineering resources are "workstations" and design tasks are "jobs" that flow among them [90]. Accordingly, a set of techniques to manipulate the design activities has been developed, such as Design Signal Flow Graphs [91] and Design Process Network [92]. Design Signal Flow Graph represents a directional diagram of relationships among a number of design tasks. The path transmission between two tasks is defined as the product of all branch transmissions along a single path and is used to calculate branch transmission cost. Design Process Network presents a uniform representation scheme for the design process entities which convert the design information into knowledge. This representation scheme is developed by recognizing all design entities and their input-output relationships. Similar to Design Signal Flow and Design Process Network, as recent researches of Complex Systems deal with the dependency issue in a design process, the last few years have witnessed a resurgence of interest in complex systems [93, 94]. Research progresses have already been made, such as Design Structure Matrix method (DSM) [95], Design Process Decomposition method [96], Petri-net based method [97] and Project Task Coordination model [98]. Design Process Structure reveals the dependency relationships of design tasks using a binary matrix, and by matrix operations designers reduce the unnecessary design iterations. Petri-net model includes the concepts of events as the intermediate connections between tasks. These two models are frequently applied in recent research. Although different approaches focus on different aspects of design process, one common principle that does not change is maintaining the independence of design tasks thus decreasing the requirements for collaboration. Comparably, in this dissertation, it has been assumed that design collaboration cannot be completely avoided and effective collaboration is at least equally important as a management of task dependency.

2.4.3 Researches on Product Data Management

In this group, design process is defined as the management of the product data for different disciplines in different design stages. During this process, the technical, scientific, and interdisciplinary dependencies of the information could be established and maintained to support processing of various types of design data [99, 100]. The key issue for properly managing product data is data consistency, which maintains the data integrity through application of data operation and transaction control algorithms. Applications of research on product data management can be observed in the success of several commercial software releases. Among them, from our software survey Windchill from PTC, TeamCenter from Unigraphics and Enovia from IBM earn more reputations in the market.

An often discussed drawback of product data management software is that current software has been proved to be insufficient to support the representation and exchange of major engineering information in design. Except for geometric modeling, other engineering information, such as evaluation, analysis, simulation, etc., which are also critical to support basic design information sharing among the network, are not precisely depicted as object-oriented models. To seamlessly connect engineers, Senin, Pahng and Wallace proposed their work, Distributed Object Modeling & Evaluation (DOME) that attempts to share engineering views between members in teams or disciplines [101]. DOME framework extends the ordinary information range of product data management. However, for supporting multi-user design cooperation, sharing engineering information is an initial step; some cooperative design activities, such as cooperatively generating and manipulating engineering information are also essential and need to be supported in a distributed and multi-user environment.

The three groups of design process research describe the design process in different levels of abstraction. A brief comparison from Lu and Cai [102] distinguished the research of design process modeling by researchers' study disciplines and three aspects of design process [102]. In Figure 2.6, design decisions, activity manipulations and data support are mentioned as three aspects of design process. Within each aspect, the corresponding elements of design process are divided by three doted polygons. Lu and Cai [102] argued that previous research assumed design processes as pure technical activities and ignored stakeholders' social interactions. In fact, differences of individual's background and social role lead to different understandings of product design. Without a coordination of engineers who have different design understandings, there is still a risk to encounter some failures of product design, although product data management and workflow management are applied.

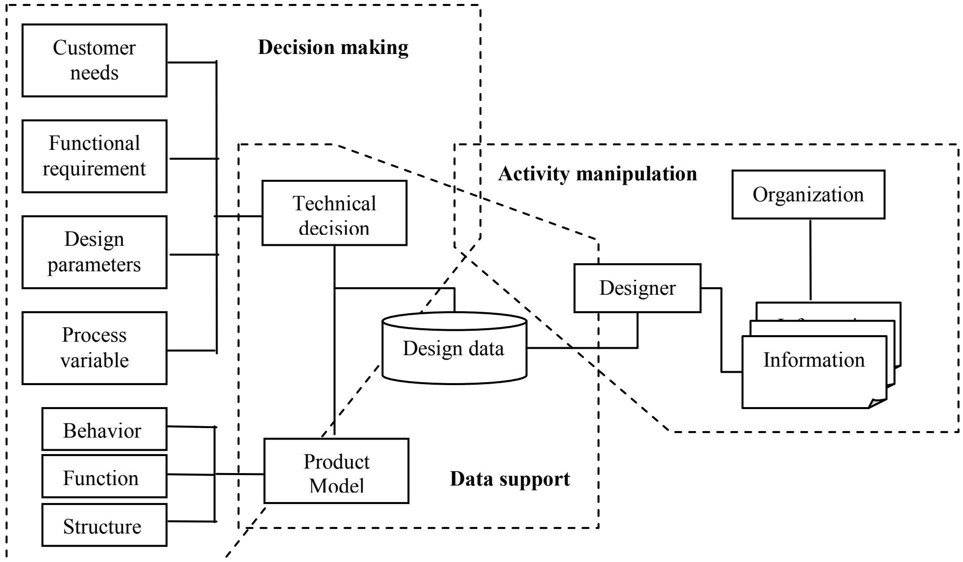


Figure 2.6: Traditional design process modeling approaches [102]

Aspects / Disciplines		Decision Making	Activity Manipulation	Data Support
Tech.	Engineering /Physics	Suh QFD Yoshikawa Pahl&Beitz		
	Operation Management		Adler Eppinger Bras Bar-Yam Yassine Zhao	
	CS/Data Management			Majumder Sriram. Krishnamurthy Pahng
Social	Decision Theory /Cognition	Hazelrigg Mistree		
	Organization Theory			
	Social Construction			

Table 2.1: Design process modeling approaches (adapted from [102])

Lu and Cai presented a Socio-technical idea to clarify the relationships of various social and technical influential factors in the design process.

2.4.4 Why Design Process Modeling

The research in design process modeling area is to help engineers understand mechanical design. Ullman have proposed their models of mechanical design in their research [103]. These research describe mechanical design in various aspects, abstract levels and reveal many important elements in design. Following the philosophies in the research of design process modeling, an engineer knows more about the overall of mechanical design, its characteristic and its stages.

The research of design processing modeling is used to build a distributed mechanical design framework in Chapter 3. The research works in Chapter 3 is to find the important elements that are required to achieve design collaboration in a distributed mechanical design. Compared with traditional design process modeling techniques, our focus is to find the important research issues in distributed mechanical design that need to be discussed in the research of this dissertation.

2.5 Online Engineering Tools and Group Design System

Online engineering tools and Group Design System (GDS) are the software, that support engineers in different locations work together through computer network. Some of the tools include internet-based distributed collaborative engineering analysis [104], agent based collaborative Design for X [105], multi-client collaborative shape design system CADAC with server-based geometry kernel [106], and multi-user modeling of NURBS-based objects [107]. Among these tools, internet-based distributed collaborative engineering analysis provides an engineering analysis framework that allows remote users to conduct finite element analysis, including pre and post processing, collaboratively over the internet. Agent based collaborative Design for X reported a case study of distributed web applications using CyberCo [105]. The key purpose of this case study demonstration is to extend the knowledge and insights into this emerging field where an increasing number of web applications are developed and deployed for collaborative product development and realization projects. Multi-client collaborative shape design system and multi-user modeling of NURBS-based objects proposed two geometric modeling tools for a multi-user environment [107].

2.5.1 Why Online Engineering Tools and Group Design System

Since distributed design changes the design scenario, the development of design tools and GDS are required to support various engineering design activities for an engineering team. Currently not many design activities can be supported by existing tools or design systems. New tools need to be developed by researchers and commercial software companies.

The application of powerful design systems and tools can short design time, save development cost and increase product development companies' market competition. Besides the traditional CAD/CAE/CAM software, software companies have recognized the importance of some special design system and tools such as collaborative design tools, product data management system, Bill of Material management, etc. There is still a big gap between engineers' need and the functions of developed software.

CHAPTER 3
DISTRIBUTED MECHANICAL DESIGN FRAMEWORK

The global competition in product development is requiring enterprises to shorten their product development time, reduce cost, and improve the quality of the designed products. In order to achieve these goals, engineering teams located in different places are required to be involved into the product developments, contribute their efforts, and collaborate with each other. In this chapter a discussion of the constituents needed to achieve successful collaboration between engineering teams based on computer network are presented. The description of these constituents provides an overall picture of design collaboration which is very important in a distributed product development environment.

Using distributed mechanical design, product development is implemented by multiple engineering teams. Each team has its design considerations and works with other teams simultaneously. In most cases, the design decisions of one team effects the decision of another. This is especially true when different aspects are considered for the same components and systems. There is a complex interdependency relationship between the design decisions of different engineering teams in different design stages. For example, in a product development, product cost, product quality and product maintenance are some of the design considerations. Engineering team responsible for product cost cannot make their design decisions just based on cost information. The design considerations of other engineering teams cannot be isolated and neglected. To achieve successful design collaboration, coordination is required to connect the design decision makings among multiple teams.

Besides the interaction of coordination, there are other interactions that are important to achieve successful design collaboration. One type of interaction is classified as synchronization, which keeps the product data or any engineering information consistent in all teams. Synchronization is obtained in current computer system using network data transferring. The other type of interaction is classified as communication which also uses computer network to exchange the product and process information between multiple teams. The two types of interactions synchronization and communication are embedded in the different design support systems (see Chapter 7 for more details).

In this dissertation, design collaboration is defined as three types of interactions between engineering teams, coordination, synchronization and communication. In order to briefly describe the distributed mechanical design, an introduction to design collaboration is given in Section 3.1 and a comprehensive distributed mechanical design framework is presented in Section 3.2, which includes major elements of distributed design and reveals the relationships of these elements. In this chapter, the focus is on answering Research Question 1 (Section 1.2). Design collaboration is treated as the key to successful distributed mechanical design. The rest of the chapters in this dissertation Chapter 4, 5, 6, 7 provide approaches or design systems that support engineers to realize different elements described in the distributed mechanical design framework.

3.1 Design Collaboration

Design collaboration has its intrinsic characteristics, it is complex and influenced by not only the technical aspects, but also the social aspects of the engineers in the design environment. Therefore design collaboration must be described as Socio-technical interactions, as shown in Figure 3.1. The lower plane, which includes engineering teams and their roles, represents the design environment, the infrastructure in which a specific design campaign is to take place. In the middle of the lower and upper plane, two arrows represent social and technical activities of engineers when they fulfill the responsibilities of their roles. The left upper plane shows the technical activities including design decision and design operations involved in product design data generation during a specific design campaign within the environment. In this research, two types of technical activities are considered - design decision and design operation. Design decision is the activity that an engineer or a team determines as a part of product design by selecting the suitable design solution from a range of options. Design operation is the activity, in which an engineer or team converts the design decisions to some types of entities, such as geometric modeling, simulation modeling, and fast prototyping. The right upper plane shows the social activities of an engineer or team. These social activities include design data, process, and resource management, which are important constituents of distributed mechanical design. The interactions of social activities are communication, which are exchange of various information such as data, command, knowledge, and experience.

Initiated by a design objective, the design campaign activates the social and technical roles of engineers or teams, and drives various social and technical activities and outcomes the product data in the left upper plane. The co-construction, which occurs in the upper planes, is a combination of Socio-technical activities and their interactions.

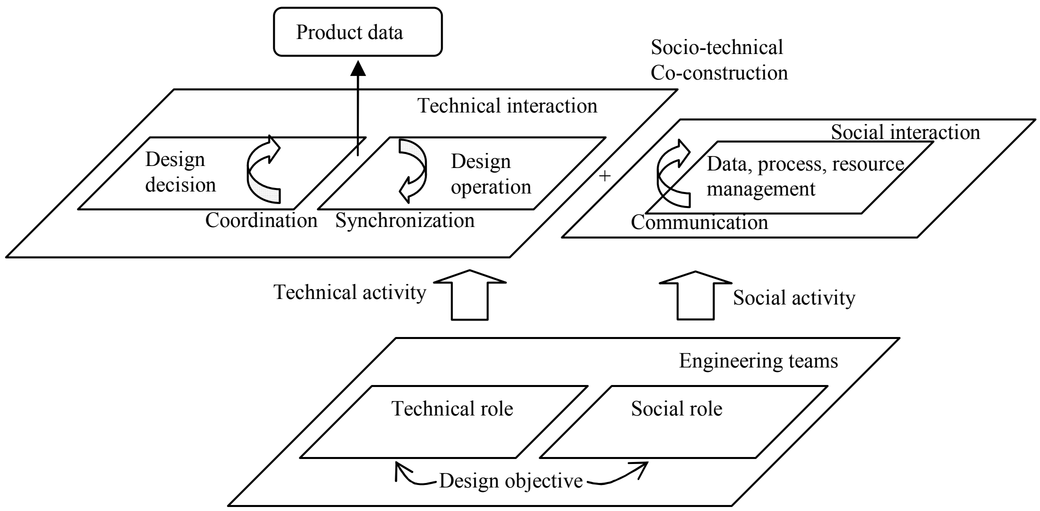


Figure 2.1: Socio-technical interactions

In Figure 3.1, technical activities can be considered as conducting various design activities, which are associated with engineers' technical roles (e.g. generating design concepts, completing detailed design, generating product data and so on). All these technical activities are related with decision making, and can be classified into two parts design decision and design operation. Social activity can be considered as conducting various management activities, which are associated with engineers' social roles (e.g. updating product data, scheduling design process, assigning design tasks, etc.). Through Socio-technical activities, product data is generated to progress product development.

Technical interactions in this dissertation include coordination and synchronization. For a design decision, coordination is essential for decisions that are not only suitable for local design requirements, but also acceptable for global product development requirements. For a design operation, synchronization is essential to update product data and other design information in real time or periodically. The social interaction in this dissertation represents communication among different design members and teams. In order to manage a design process, communication is essential to avoid misunderstanding.

3.2 Distributed Mechanical Design Framework

The Socio-technical interactions, presented in Section 3.1, are explicit descriptions of design collaboration, which are major elements in distributed mechanical design. As shown in Figure 3.2, decision coordination, data synchronization and communication are three critical elements for distributed mechanical design. Accordingly, Socio-technical activities are the elements driven by engineers' different roles. In a design campaign, engineers play both technical roles and social roles. The former is represented in the technical activities, while the latter represent social activities. The engineers' responsibilities based on Socio-technical roles are to gather product and process information. The unique considerations of each engineering team are formed and evolved when engineering teams become part of a community and begin to interact with each other on their design decisions. The product information is generated by engineering teams after design decisions are made. Since community members tend to have different formats to describe their considerations, knowledge representation is essential for capturing the technical information behind decision-making and converting them to standard format. Similarly, product model or data is used to extract and store product information. Besides knowledge representation and product data, information sharing supports multiple teams to have access to the same product or process information. Design process organization analyzes the dependency relationships of various design activities and can be used to plan or schedule a distributed product development. Design resource utilization controls all design resources (i.e., engineers, computation capacity, etc.) that are divided into different teams. Through coordination, synchronization, and communication, the framework handles the interactions in a design community. Effective interactions between engineering teams in distributed mechanical design motivate design innovation at the early design stage and prevent unnecessary design iteration at the later design stages.

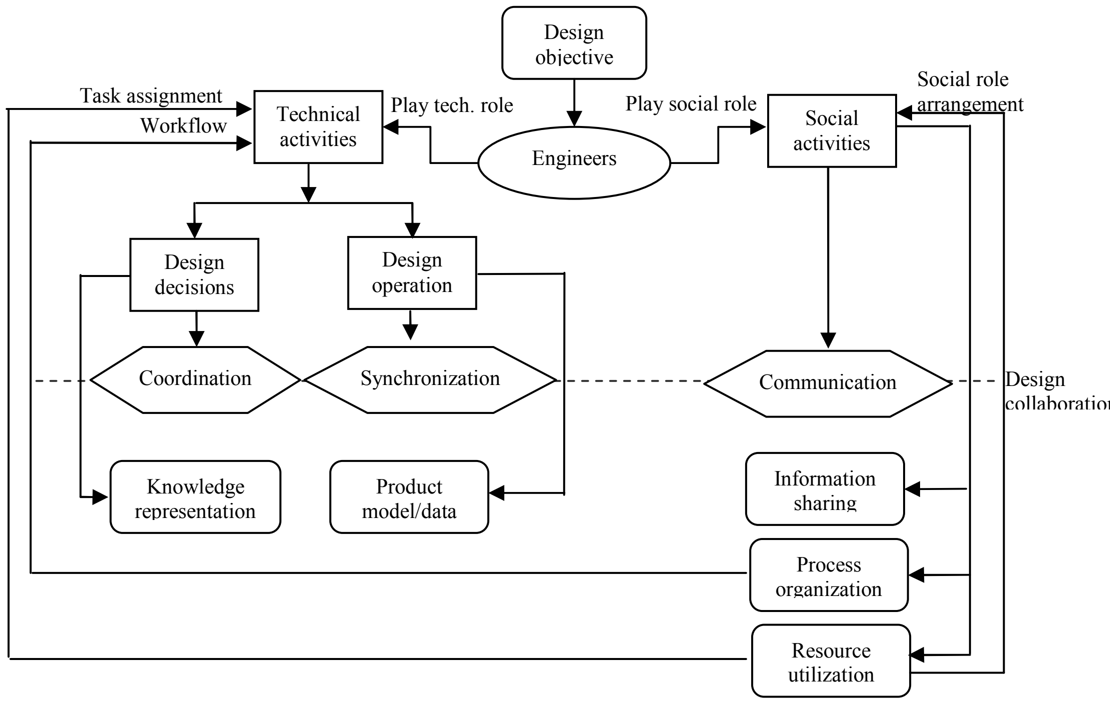


Figure 2.2: Distributed mechanical design framework

3.3 Major Elements of Distributed Design Framework

The elements of distributed mechanical design framework were presented in Section 3.2. This section offers a more detailed discussion related to the content and definition of major elements of the distributed mechanical design framework.

3.3.1 Technical Activities

In a design campaign, engineers perform both technical roles and social roles. Technical activities are conducted when stakeholders are employed in technical roles in which pure technical factors determine the activity results. The results of performing a technical activity are independent of the human factors, such as engineers' positions or social background. Customers, designers, managers, and manufacturers play various technical roles in different design situations. Accordingly, they choose different technical activities to accomplish their roles, such as searching, collecting, viewing, drawing, modeling, etc. In most cases, after conducting various technical activities, an engineer completes his/her technical role with a coordinated design decision and generated product data.

In practice, technical activities of an engineer vary as the mechanical design progresses into different design stages. Generally, the technical activities of the engineers in the distributed design include:

- Gather and analyze the customer needs based on the marketing information,
- Select conceptual design alternatives,
- Draw a function structure for the product,
- Model a product structure and performance,
- Assign values to the design parameters,
- Make design decisions,
- Test design results,
- Edit documentation, etc.

All these activities can facilitate decision making by generating information related to the product. In a multi-user environment, to make a design decision or generate product data collaboratively, systematic approaches and design tools are required. These approaches are developed to support engineers to make collaborative design decisions and these tools are designed to support distributed applications with the synchronization of data on multiple sites. In this dissertation, some of these collaborative design tools are presented in Chapter 7. These tools support multiple engineers to use real time collaboration to perform design activities.

3.3.2 Social Activities

Mechanical design also involves social roles and corresponding social activities, which are normally influenced by the organization structure, culture, individual background, and other social factors. During a mechanical design, the participants perform different social activities in accordance with their different social roles, such as planning, scheduling, managing, controlling, learning, discussing, instructing, etc. Through these social activities, engineers exchange their opinions and make consensus based on their different understandings of product and process information.

Although, for engineers, technical roles and social roles usually are combined, they take different priorities in different design stages. In the early design stage engineers spend more time on their social roles and, thus, perform more social activities, while in the late design stage they take more on technical roles. Some of the social activities of design engineers might include:

- Organize product development teams and process,
- Make a product development plan and schedule design process,

- Discuss basic concepts, layout and achieve consensus for further development,
- Inform engineers of their social and technical roles,
- Acquire knowledge and experience from others,
- Manage dynamic organization and schedule, etc.

In order to perform the social activities, in a distributed environment, a base system must be provided for engineers, which supports basic management functions for data, process, and users. In Chapter 7, an example base system, that has been implemented, has been presented. Besides a base system, to organize the design process for a distributed product development, a design process modeling approach is required to describe different design processes, select a suitable process candidate, and implement design process for product development. The design process modeling approach in this dissertation is introduced in Chapter 5.

3.3.3 Synchronization and Communication

Synchronization in this research refers to updates of working objects with any design modification. For multi-user design collaboration, the working objects must be synchronized so that all stakeholders are informed of others' design changes. Synchronization is an important form of multi-user interaction between different engineers. In this dissertation, real time synchronization requires the applications to respond to design changes within a small response time. Any synchronization which does not satisfy the requirement for response time is not real time and is classified as asynchronous collaboration. Most of data synchronization methods, such as version control and data access control, are not real time according to the definition, requiring engineers to check design changes periodically.

Communication refers to information exchanges among individuals or design teams. For engineers to perform social activities, communication is often needed f to obtain opinions of other engineers. There are many types of non real-time communication, such as email, notification, forum, etc, Also there are many real-time communication , such as video or audio conference, text chat, comment illustration, etc,. In Chapter 7, various communication tools based multimedia information is discussed including the 3D model discussion studio which is an integrated tool that supports information exchange using audio, video, whiteboard, text chat and 3D geometric model formats.

In this dissertation, the research focuses on real-time synchronization and communication, which are the two important interactions between multi-users.

3.3.4 Coordination

In distributed mechanical design, design decisions need to be coordinated to prevent conflicts and achieve proper tradeoffs. The coordination in this dissertation is defined as the involvement for various engineering considerations into decision making to find satisficing design solutions.

It should be pointed out that most conflicts in distributed mechanical design are caused when different engineering teams make design decisions in isolation. Therefore, effective coordination is indispensable to investigate the influence of each design decision. Although some research achieves efficient design by maintaining the independence of design tasks, in most of design practices design tasks are coupled with each other. The coordination is required by the design campaign to solve coupled tasks. The coordination becomes the pivot issue in the collaboration because engineers normally have limited capability to identify the influences of their decisions on others. Due to lack of relevant design information, engineers are unaware of potential conflicts, which can lead to design iterations during later design stages. In this dissertation, coordination is embedded into the collaborative design decision making approach, which is discussed in Chapter 4.

3.3.5 Information Sharing, Process Organization and Resource Utilization

Product model management consists of application of information technology to all aspects of product developments, manufacturing, and operation. A product design is accompanied with collection, creation, and management of various engineering models or data. Collecting the models and data are usually the start point of an engineer's design activities; management is the essential steps to keep models and data consistent throughout a whole design process; and solution generated from the models is the end point of a design decision.

Design resource consists of engineers, computer hardware, and software that can be used to accomplish the design activities. The design resource management is to maintain structure of engineers' organization, assign their Socio-technical roles and make arrangement for the use of design resource in different parts of product design. Process organization is a plan for design activities or events to be performed. Resource management and process organization are usually dependent on engineers' social roles. Dynamic adjustments of resource utilization and design process are often needed in a complex distributed mechanical design.

3.4 Summary

Chapter 3 a distributed mechanical design framework wa presented, which gives a picture of the ideal mechanical design for multi-user distributed environment. The framework is presented to depict major elements of a distributed design and their relations. In the framework, synchronization, communication and coordination are introduced as three major interactions between designers. In the next Chapter, the approach of the design decision making that supports distributed mechanical design depicted by the framework is presented.

CHAPTER 4
REALIZATION OF COLLABORATIVE DESIGN DECISION MAKING

In previous chapter, a description of distributed mechanical design was presented. From the description, decision making is one of the important design activities in any mechanical design. Mainly two types of design activities are mentioned - design decision and design operation. The research works in this chapter are focused on making collaborative design decisions by multiple engineers or teams. The developed systematic approach starts from the discussion of the simplest case of collaborative decision making, where two engineering teams are involved. Both teams are responsible for solving the design problems in different engineering disciplines. Following the philosophy of collaborative decision making, both teams separate their decision making activities that means although their design problems are coupled, they do not attempt to integrate the design problems. Each team solves the engineering problem separately without full involvement of the other team. After the case of two engineering teams is studied, an extension of two teams to multiple teams is presneted in Chapter 5 and 7. The chapter answers research Question 2, that was presented in Section 1.2. The systematic approach of collaborative decision making developed in this dissertation is fundamental to realize distributed mechanical design.

To separate design decisions and solve coupled design problems, representation and exchange of design decision information are essential which helps avoid unnecessary design iterations. In section 4.1, the digital design interface is presented which includes a c-DSP design activity template and a design solution template. In this dissertation, it is assumed that major design activities can be represented using c-DSP design activity template which makes c-DSP a uniform format in design information exchange. The design activity template and solution template support the information flow in a design process. Activity template provides an option to describe the design problem so that engineers know exactly what kind of design decision is preferable. Solution template provides an option to describe the design solution. It is used to provide design decision information to other engineers. In Section 4.2, game theory protocols are applied to categorize the types of design collaboration strategies in distributed mechanical design. Separate design activities are constructed according to the game theory protocols. The last section, Section 4.3, introduces the approach of managing and delivering design freedom from one engineering team to the other. In mechanical design, it is essential to give engineers a feasible design spaces so that they can choose the design solutions that best satisfy the design problems.

4.1 Partitioning Product Design with Digital Interface

In order to accomplish a product realization process, it is necessary to partition it into a set of design activities. Following the philosophy of DBD, design decisions are an important type of design activity and it is necessary to represent the design information related with design decisions using the activity templates. Information flow in product realization process can be categorized into product information and design decision information. Product information describes the physical states of the product while decision information is derived from product information for directly facilitating team's decision making. Product information representation is a complex research area for which various information models and structures are presented, such as STEP developed by International Organization for Standardization (ISO). It is developed to enable the exchange of product model data between different engineering software in a product realization system or the sharing of that data by different software. Product information representation is not the main concern in this dissertation, instead we focus on design decision information. In traditional product realization process, engineering teams' decisions are not explicitly represented. In most cases, design decision information is embedded within the product data files or directly transferred between teams in the form of commands. The result is a team's design decision information, including the requirements, design intentions, objectives, constraints, tradeoff strategies, and design solution may be misinterpreted by other teams. It is necessary to present a standard data format that can be used to transfer design decision information among engineering teams. In this dissertation, design activity is referred to as the requirements, goals, and constraints that are required for design decision. Design activity template is the format used to represent the design activity. Design solution is referred to the results of the design decision, which is presented through the values of each design variables.

In this chapter, a simple two discipline example is selected to illustrate the design decision information flow. In the example, there are two engineering teams. Thermal design team and stress design team make their design decisions to select the values for two design variables C and D. The same example is used through all sections in this chapter. The example has been adapted from [108]. For the thermal design team, the design variable is the chip thickness $C \in [0.5, 0.9]$ mm, design goal is the maximum temperature on the chip is as low as possible and the design constraint is that maximum temperature $T < 70$℃. The team is to select a suitable value for variable C.

For the stress design team, the design variable is die attachment thickness, $D \in [0.03, 0.07]$mm, design goal is that maximum von Mises stress on the interface is as small as possible, and the constraint is that the maximum von Mises stress $S < 190$ MPa. The design team is to select a suitable value for D.

In Figure 4.1, the design decision activity is partitioned into as two design activities - stress design and thermal design. The information flow between the two design activities are illustrated using arrows. Based on the information flow, design activities can be separated and design decision information can be shared by the two design teams.

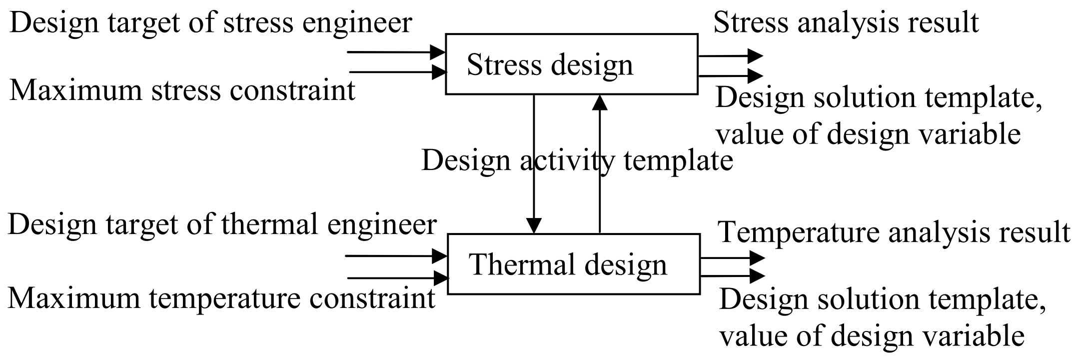


Figure 4.1: Design decision information flow

In the example, the engineering analyses are performed using different values for design variables C and D as input. The analysis results are used to generate the Response Surface which is an approximate equation that can conveniently estimate the analysis results in stress and thermal designs. Since in this dissertation the focus is not on how to perform engineering analysis, but rather on the collaboration aspects of the design, the result of analyses is given in Table 4.1. The generated quadratic Response Surface equations are also provided.

C (mm)	D (mm)	S (MPa)	T (℃)
0.5	0.03	263.612	54.210
0.7	0.03	233.018	57.290
0.9	0.03	251.139	75.279
0.5	0.05	244.778	59.859
0.7	0.05	195.887	63.562
0.9	0.05	185.698	75.218
0.5	0.07	223.258	69.767
0.7	0.07	174.194	67.536
0.9	0.07	158.356	75.274
$S(C,D)=355-113.7C-1599.7D+502.7(C-0.7)^2-3276.8(C-0.7)(D-0.05)+21187.9(D-0.05)^2$			
$T(C,D)=27.4+35C+215D+136.8(C-0.7)^2-972.6(C-0.7)(D-0.05)+866.5(D-0.05)^2$			
$BRC_S(C)=0.07$			
$BRC_T(D)=0.39455+3.555D$			

Table 4.1: FEA results and equations of stress and thermal analysis

4.1.1 Design Activity Template

An information medium between activities must be capable of capturing the design decision information into a concise, standard and disciplinary independent format. Conventionally design decision information is exchanged in the form of data and commands. In a distributed environment this conventional information communication method becomes more difficult and causes design iterations. The digital design interface in this dissertation is used to support information flow in a design process. The benefit of applying templates is that it greatly simplifies the information communication standard format will eliminate, if not reduce, misrepresentation and misunderstanding of design decisions.

In this section, a c-DSP design activity template is presented which provides a method to describe the design problem so that engineers know exactly what kind of design decision is preferable. The activity template presented in this section works as an "information package" that is transferred between multiple teams. In this dissertation design activity template is distinguished from a representation of product information or knowledge. It is a medium for representing the information that is needed in a design decision activity. Using activity template, design problem that needs to be solved in a design activity is explicitly represented.

In this dissertation is a c-DSP formulation that is employed to describe design goals, design constraints, and various design considerations. In our product realization scenario, a design activity template is an instantiation of the c-DSP. The mathematical formulation of c-DSP is shown in Chapter 2 Figure 2.1. A team's design decision activity consists of a design space and design intentions. Design space is bounded by the limit values of system variables and constraints. Design intentions are the design objectives and their tradeoff strategies represented by goals and deviation functions in c-DSPs. The dependence relationships between activities are represented as coupled variables in c-DSPs.

In the design example of this chapter, engineers in thermal and stress design disciplines represent their design activities using c-DSPs (see Figure 4.2). Engineering tools are employed to provide the stress and thermal analysis results, which are embedded into the design activity templates as the input information to find proper design solutions. The two engineering analysis results are the functions of the two design variables chip thickness C and die attached thickness D. Although the accurate result of the function for a certain set of values of design variables can only be obtained by performing analysis using the engineering tools, it is possible for engineers to estimate the analysis result using Response Surface Method (RSM) which is generated by running the analysis for a number of times and based on the obtained results to predict the results of analysis for some unknown input. In this example, the Response Surface is generated using the software "Minitab", a statistical software that can generate quadratic equations to approximate some complicated mathematic functions or experimental results.

Stress design Activity	Thermal design Activity
Given	Given
Package design and material	Package design and material
Find	Find
Die attach thickness D, Chip thickness C, d_S^-, d_S^+	Chip thickness C, Die attach thickness D, d_T^-, d_T^+
Constraint	Constraint
$C \in [0.5,0.9]$	$C \in [0.5,0.9]$
$D \in [0.03,0.07]$	$D \in [0.03,0.07]$
Satisfy	Satisfy
$S(C, D)/190+d_S^- - d_S^+ =1$	$T(C, D)/70+d_T^- - d_T^+ =1$
Minimize	Minimize
$f_s(d_S^-, d_S^+)$	$f_T(d_T^-, d_T^+)$

Figure 4.2: Stress and thermal design activity templates

Design activity template contains sufficient design information for engineers in different disciplines to understand other's design activity. It is not necessary for each engineer to have the knowledge and software skills of all disciplines, rather to let cooperative engineers know the influence and trade off of their design decisions.

In the example, the digital design interfaces between two design decision activities are shown in Figure 4.3. Stress template is the digital interface from stress team to thermal team and thermal template is the interface from thermal team to stress team. The design problem templates facilitate the information flow between different engineering teams.

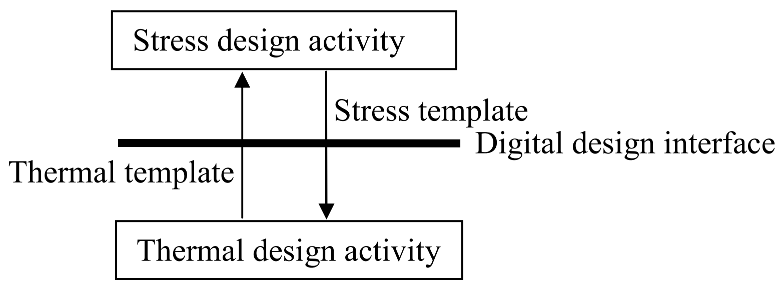


Figure 4.3: Digital design interface

4.1.2 Design Solution Template

Besides the design activity information, another type of information that is related with design decision is design solution. Design solution is the result of design decision and is usually transferred from one engineering team to its sequential teams. The sequential teams accept the design solution if conflicts are not found and continue their design activities. In this case, the design solution information is required to be exchanged between engineering teams.

Similar with the exchange of design activity information, a standard and concise format is needed to represent design solution information. This format is capable of capturing the information in various design solutions and implementable by computer software. In this dissertation, design solution template is the standard format that is used to exchange the design solution information between teams which contains various types of information including a selection of some design options, a preference to some alternatives and a guidance for other engineers in their decisions. The solution template is a general format that can be used to represent these different types of design decision information.

In this dissertation, three parts in the solution template are used to represent the information in a design solution. These parts are feasible solution point, feasible solution range or set, and solution response. The above specific information in a design solution is represented as a feasible design point. By seeking a feasible design point, all values of design variables are selected. The relevant part of product design is completed with a design solution. The design variables whose values are decided in a design solution are classified as master design variables for the design activity. The preference information in a design solution is represented as a feasible solution range or solution point set. The feasible range or set indicates that some design decisions are preferable for engineers and design space is narrowed before final solution point can be found. The design variables whose values are not fixed in a design solution are classified as slave design variables for the design activity. The guidance information in a design solution is represented as response information which describes the response of an engineering decision if the different values for design variables are selected. In this dissertation, the response information in a design solution can be categorized into two types, Response Surface and Rational Response Set (RRS). The Response Surface is a collection of data that reveals a disciplinary performance varies with the values of design variables. The RRS is also a collection of data. It indicates the values of certain set of design variables vary with another set of design variables.

The design solution template is illustrated in Figure 4.4. In most cases, only one of two parts needs to be provided. In the next section, according to the different design collaboration strategies used to construct a design game, the corresponding information in a design solution template is also different.

Given
A design activity template
Design master variables x_j, j=1….m
Design slave variables x_j, j=m….n
Find
Solution point
x_j, j=1…n

Solution point set
x_j, j=m…n

Solution range
$x_j \in [x_j^{min}, x_j^{max}]$, j=m….n

Solution response
$RS(x_j)$ or $RRS(x_j)$, j=1….n

Figure 4.4: Design solution template

In the design example of this chapter, engineers in thermal and stress design disciplines represent their design solution using solution templates. The design solution template is illustrated in Figure 4.5. In the example, stress design team is assumed to transfer its design solution template to the thermal team. The stress design team has the privilege to choose the value for design variable C and pass the optional values of design variable D to thermal team.

Given
Stress design activity template
Design master variable C
Design slave variable D

Find
Solution point
C

Solution set
D

Solution range
$D \in [D^{min}, D^{max}]$

Figure 4.5: Design solution template of stress design

The design solution template in this dissertation is a type of digital design interface exchanged from various engineering teams. The design solution template facilitates the information flow and makes it possible for engineering teams to understand each other's decision better.

4.2 Game Constructs for Collaborative Decision Making

Game theoretical principles facilitate construction of design collaboration in product realization. The basic concepts of game based design and generic game construct have been introduced in Chapter 2. In this section, three available design collaboration strategies are presented which categorizes the collaboration between the teams of two engineering disciplines.

In traditional mechanical design, trial and error decision making process is often used. Trial and error decision making is an easy approach to avoid frequent design collaboration among various engineering teams. In a trial and error process, engineering teams make their design decision independently without any cooperation from other teams. The feasibility of the design decision is largely dependent on engineers' design knowledge and experience. From the perspective of game based design, trial and error process is a special Nash game. In a Nash game, engineering teams make design decisions and generate a set of design solutions. The Nash solution is the intersection of design solutions from these teams. In a trial and error process, engineering teams do not cooperate with each other. They generate one design solution and expect that it is an acceptable design solution for other teams.

A more general form of trial and error design collaboration strategy is Nash noncooperation game protocol, in which the engineering teams do not receive other teams' c-DSP design activity templates. The design decisions are made without exchanging design decision information. In order to find a solution that is acceptable for multiple teams, each team constructs a Best Reply Correspondence (BRC) or Rational Response Set (RRS) which includes a set of acceptable solutions or an acceptable range of design solutions. Intersection of all these RRS or BRC is expected to find the final solution. The design scenario of using noncooperative game protocol to solve chip package example is represented using the following formulation:

Find $BRC_T \times BRC_S$

In a Nash game, each engineering teams generate a feasible BRC solution sets or solution range with the selection of a set of values for certain design variables. As an example, for the chip design, stress design team finds the proper values of design variable D assuming that a set of values are selected for the design variable C. The generated BRC_S is a corresponding relationship of two design variables, $D=BRC_S(C)$. If the thermal design team also generates its BRC_T, which is a corresponding relationship for design variables C and D, $C= BRC_T(D)$, BRC_T and BRC_T both are generated to represent a set of acceptable design solutions. And the intersection of these two sets of design solutions is a possible solution for coupled design activities. The solution obtained from above Nash game is mathematically a Nash solution because this solution satisfies the following condition: a pair (C^N, D^N) is a Nash solution if: [54]

$$f_S(C^N, D^N) = \min f_S(C^N, D) \text{ and}$$
$$f_T(C^N, D^N) = \min f_T(C, D^N)$$

A Nash noncooperative protocol game is an ideal design scenario, where no design iterations are needed. But the process of generating BRCs can be a time consuming process. In some cases, noncooperative game is not easy to implement.

Another possible design scenario is based on full cooperation game protocol Pareto. A typical case of full cooperation is concurrent design or integrated design, where the different engineering teams are completely informed of other teams' design activities. All teams have full access to the information about others' decision making. In this dissertation, design activity template and solution template are used to support information exchange between teams. The result of a full cooperation is solved by combining players' c-DSPs. Design scenario using Pareto cooperative game protocol. As an example, to solve chip package design problem it will be represented using the following formulation:

Minimize $f=W_S f_S+W_T f_T$

In a Pareto game, engineering teams receive design information about their partners. Based on the information, engineering teams combine the design activities and add other teams' consideration into the design decision. In the chip package design example, a possible way of constructing Pareto game is to pass the Response Surface information of stress design team to thermal team and then let thermal team to select the values for design variables C and D. The final solution is mathematically a Pareto solution because it will satisfy the following condition: a pair (C^P, D^P) is Pareto optimal if no other pair (C, D) exists such that: [54]

$$f_S(C,D) < f_S(C^P, D^P) \text{ and}$$
$$f_T(C,D) < f_T(C^P, D^P)$$

Following leader/follower game protocol, the leader solves c-DSP using follower's BRC information and the follower accepts the instructions from the leader as design rules. Using the same design example, a possible way of constructing leader/follower game is to assume that stress team plays the role of leader and thermal team plays the role of follower. Design scenario of using leader/follower game protocol to solve chip package example is represented using the following formulation:

Minimize $f_S(BRC_T)$

Stress team calculates chip thickness design variable C based on the RRS_T or BRC_T information provided by thermal team. Stress team uses the thermal team's RRS to predict thermal team's response on a design decision and finally find a solution that considers the product performance for the thermal discipline. The follower receives a design rule from the leader, which is usually a feasible solution set or solution range, to make his/her decision to accomplish the design. The final solution is mathematically a Stackelberg leader/follower solution because it satisfies the following condition: a pair (C^S, D^S) is Stackelberg solution when the leader specifies the rule (C_k, D_k), k=1…n which is a set of optional values of variables C and D for the follower and acceptable for the leader in minimizing $f_S(C, D)$ and follower finds (C^s, D^s) if no other pair (C, D) exists such that: [54]

$$(C^S, D^S) \in (C_k, D_k), k = 1 \ldots n$$
$$f_T(C, D) < f_T(C^S, D^S)$$

Since leader's decisions are made before that of follower's, the leader in this game has more freedom to explore the design space and therefore can ensure superior result from his/her discipline. A leader/follower game protocol facilitates collaborative decision making without iteration. In this dissertation, follower's BRC can be described using mathematical equations, approximate Response Surface or a set of solution points. All these information is supported by design solution template.

4.2.1 Implementation of Game Protocols

In this section, the digital interface and game construct information that are discussed in the previous two sections are used to solve the chip package design. Engineering analyses results are generated to simulate a complete scenario to accomplish collaborative decision making based on various design collaboration strategies. The engineering data of the design example was provided in Section 4.1 in which function S represents stress and T represents temperature and is referred below:

$$S(C,D)=355-113.7C-1599.7D+502.7(C-0.7)^2-3276.8(C-0.7)(D-0.05)+21187.9(D-0.05)^2$$

$$T(C,D)=27.4+35C+215D+136.8(C-0.7)^2-972.6(C-0.7)(D-0.05)+866.5(D-0.05)^2$$

4.2.1.1 Implementation of Trial and Error Design Process

Following the discussion in the pervious section, we consider the scenario where the stress design team is responsible to choose a value for design variable D, run the stress design activity template on computer and solve the maximum stress. The value of design variable D is adjusted based on the stress analysis result and passed to the thermal design team, who runs the thermal design problem template, calculates the maximum temperature and determines a value of C. The process is repeated until both teams converege to a solution..

Solving a coupled decision making problem using the trial and error process is a simulation of traditional product development process. The number of iteration depends on the initial values and problem itself. Teams cannot control the result. If stress team assumes C=0.9mm and runs the stress design activity template, the design solution is S=156.13MPa when D=0.07mm. After the value of design variable D is passed to the thermal team, the value of design variable D is updated by running thermal design activity template. In the best case, the results of the two teams converge after a design iteration.

Stress design: C=0.9, → S=156.1285, D=0.07

Thermal design: D=0.07, → T=66.7713, C=0.6434

Stress design: C=0.6434, → S=183.6263, D=0.07

The final result solved using trial and error approach is C=0.6434mm, D=0.07mm, S=183.63MPa, T=66.77

4.2.1.2 Implementation of Pareto Game Protocol

The implementation of Pareto cooperative game protocol to achieve collaborative design decision is straightforward. A combined c-DSPs design activity is formed and solved. The cooperative game of the design example is illustrated in Figure 4.6.

Given
Package design and material
Find
Chip thickness C, d_T^-, d_T^+
Die attach thickness D, d_S^-, d_S^+
Satisfy
$S(C, D)/190+d_S^- - d_S^+=1$
$T(C, D)/70+d_T^- - d_T^+=1$
Minimize
$f= 0.5[d_T^-, d_T^+]+0.5[d_S^-,d_S^+]$

Figure 4.6: Combined design activity

It must be noted that the tradeoff strategy between the cooperative teams is represented as the deviation function in the combined c-DSP. It depends on the teams to assign the weight of each design goal, which is a human task suffered from the bias and lack of information. Some research is focused on providing a mathematic approach to help select the proper value of weights. For simplicity, all goals are assigned the same weight. The result of the cooperative game is: C=0.07mm, D=0.7785mm, S=160.9MPa, T=69.27℃

4.2.1.3 Implementation of Nash Game Protocol

Nash noncooperative game protocol is implemented to determine the intersections of BRC generated by multiple teams. In a Nash game, each engineering team is assigned a set of design variables for their design activity. The values of these variables are selected by the team. Since the engineering team only has the privilege to control the values of a part of the design variables that are used in its design activity, the values of the rest of design variables cannot be controlled by the engineering team and are selected using Design of Experiments (DOE) techniques such as full factorial approach. The engineering team solves its design activity based on the different values of design variables which are not controlled by the team and find its BRC which is the optimal values of design variables that can be controlled by the team. The BRC can be generated in the form of solution set, solution range or Response Surface. Since Design of Experiment data is collected by solving the c-DSPs, there is no random error or measuring error like physical experiment. But the selection of experimental points is still a DOE process and statistical principles can still be applied to the data generated. The two engineering teams construct their BRC concurrently without additional information exchange.

In the design example, the stress design team carries out experiments by evenly changing the value of design variable C and the output result is:

C=0.5, $\rightarrow$ D has no result

C=0.6, $\rightarrow$ D has no result

C=0.7, $\rightarrow$ D=0.07

C=0.8, $\rightarrow$ D=0.07

C=0.9, $\rightarrow$ D=0.07

From above result, it is obvious that the BRC of the stress design team is D=0.07. Concurrently thermal design team carries out experiments and the output result is:

D=0.03, $\rightarrow$ C=0.5012

D=0.04, → C=0.5367

D=0.05, → C=0.5723

D=0.06, → C=0.6078

D=0.07, → C=0.6434

The BRC of the thermal design team is C=0.39455+3.555D. The intersection of the two BRC gives the Nash solution for this design example. The result of Nash noncooperative game is: D=0.07mm, C=0.6434mm, S=183.63MPa, T=66.77℃

4.2.1.4 Implementation of Leader/follower Game Protocol

In the leader/follower game, follower team constructs BRC for the leader. The leader team makes decision using the BRC from the follower and the design activity template. The BRC generation approach is the same approach used in Nash game. The follower engineering team solves its design activity based on the different values of design variables which are not controlled by the team and find its BRC which is the optimal values of design variables that can be controlled by the team. DOE techniques are employed to select values of input design variables to generate BRC. For the purpose of increasing the accuracy of the follower's BRC, the follower uses the Response Surface Method to find an approximate function to replace BRC data. When leader team solves its design activity, follower team's BRC is used to calculate the values of design variables that are controlled by the follower team. Since the coupled variables in leader team's c-DSP are now replaced by design variables that are controlled by the leader team, the leader can solve the c-DSP and find feasible design solution set or solution range. The feasible solution set or solution range is passed to the follower team as a design rule. The follower attempts to solve the design activity within the solution set or range specified by the leader team.

In the design example, if stress design team is the leader, the stress design activity template is solved by adding BRC of thermal design activity, C=0.39455+3.555D, the design solution of the stress activity is C=0.6434mm and D=0.07mm which is an optimal solution that the follower team can accept. If thermal design team is the leader, the BRC of stress design activity is constructed as D=0.07mm, and the value of design variable C is selected as C=0.6434 to optimize thermal design activity. After design solution C=0.6434 and D=0.07 is passed to the follower stress team, the final solution is acceptable for thermal design team. No mater which teams plays the role of leader, the result of the leader/follower game is the same: D=0.07mm, C=0.6434mm, S=183.63MPa, T=66.77℃

In the above implementations of leader/follower protocol, leader delivers a design solution point to the follower and let the follower to decide whether the solution point is acceptable or not. In a more general implementation, leader delivers a set of solution points or a range of design solutions and let the follower have the design freedom to select the acceptable design solution from alternatives. To find a proper solution range or solution points, c-DSP technique needs to be extended to include the consideration of the robustness of design solution to variation.

4.3 Maintaining Design Freedom in Leader/follower Design Strategy

In Section 4.2, three game based design protocols were discussed. Among them, leader/follower protocol plays a very important role in distributed mechanical design. Through applying leader/follower protocol different part of design activities can be linked together and design decisions can be made collaboratively without iterations. An issue of constructing leader/follower game in mechanical design is that current optimization approach can only find a design solution point for an engineering problem. In a leader/follower game, more generally the leader needs to find a solution set or solution range that can be delivered to pass design freedom from leader to follower.

The research work in this section is to introduce a reformulated c-DSP that can be used to find allowable design solution range or solution set. The objective is to add new considerations into the original c-DSP deviation function. Compared with other research on collaborative decision making, the reformulated c-DSP is developed to provide the leader with a control of the design freedom of the follower.

4.3.1 Robust Design Approach of Achieving Design Flexibility

In this dissertation, Taguchi's robust design method is used to maintain design freedom in a leader/follower game. Taguchi's robust design method has been widely accepted to improve design quality of products and processes. [109-111]. Whereas various other approaches assume that a good design meets a set of well defined functional, technical performance and cost goals, Taguchi states that a good design minimized the quality loss over the life of the design where quality loss is defined to be the deviation from the desired performance.

While majority of the early applications of robust design consider manufacturing as the cause for performance variations, recent development in design methodology have produced design approaches and methods that introduce the robustness of design decisions [112,113]. In Chang's work, Taguchi' parameter design concept is used to support teams in communicating about sets of possibilities and make decisions that are robust against variations in the part of the designs done by other team members. In their model the uncertainties between different teams are modeled as noise factors. A part of the robust design applications [113] are to apply the robust design concept to the early stages of design for making decisions that are robust to the changes of downstream design considerations (called type I robust design). Furthermore, the robust design concept is extended to make decisions that are flexible to be allowed to vary within a range (called type II robust design) [112]. In this dissertation, the Type II robust design is discussed, in which performance variations are contributed by the deviation of control factors rather than the noise factor. In Figure 4.7, design decisions are isolated into different design stages. To avoid major design changes, it is necessary to make a design decision robust to the design variation so that consequent activities has enough design freedom to make design decision.

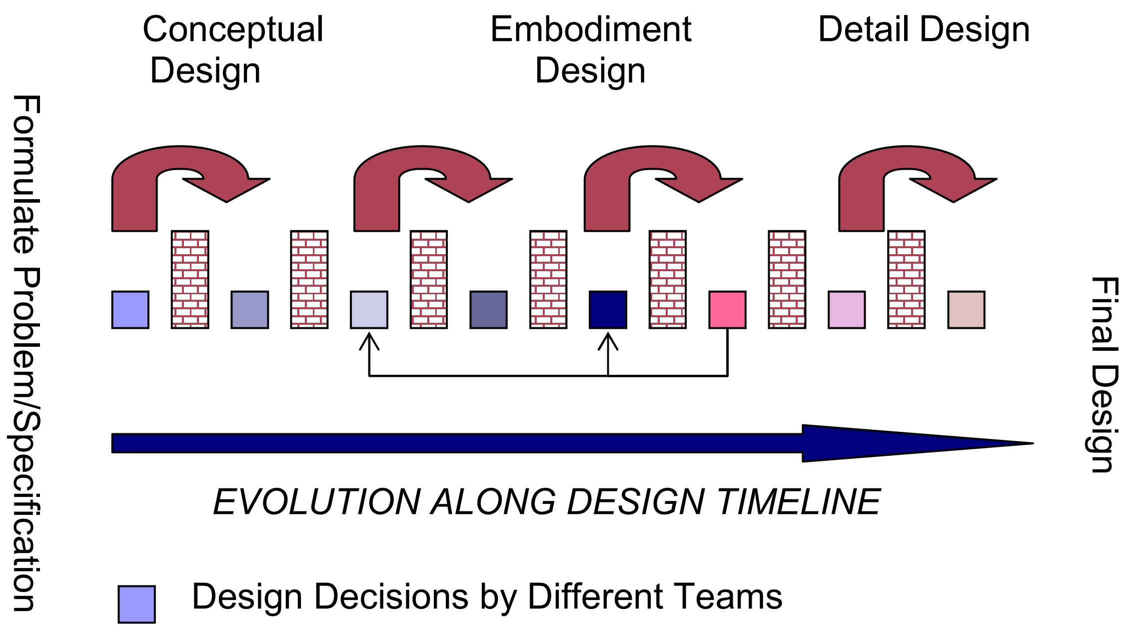


Figure 4.7: Type II robust design approach

The concept behind Type II robust design for searching a flexible design solution is represented in Figure 4.8. For purpose of illustration, assume that the performance y is a function of variable x. Generally in this type of robust design to reduce the variation of response caused by variations of design variables, instead of seeking the optimal value a designer is interested in identifying the flat part of a curve near the performance target. If the objective is to move the performance function towards its target and if a robust design is not sought, then the optimal solution is chosen. However for a robust design, the robust solution located at the flat part of function curve is a better choice if the variation of the performance function at robust solution is much smaller than that at optimal solution, at the same time the means of the function at two solutions are close.

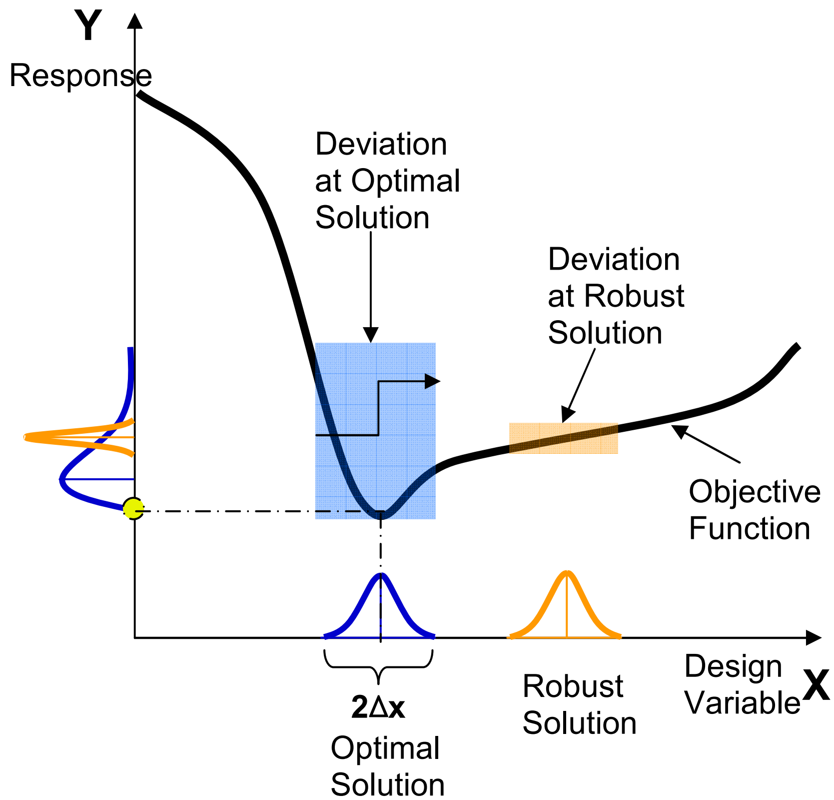


Figure 4.8: Robust design solution

In distributed mechanical design, robust design approach can be used for a leader team to find a solution range that can deliver design freedom to a follower team. A research work of using robust design to increase design flexibility is from [57]. In the research, Chen and Lewis presented an optimization formulation that takes the deviation of objective function into consideration. In Figure 4.9, Chen and Lewis's formulation for achieving design flexibility is given. The formulation has a minimization function which combines the need to minimize the mean value and standard deviation of the objective function.

Given

System design variable deviation, Δx_i

Find

System design variables, x_i $i = 1, \ldots, n$

Satisfy

System constraints (linear, nonlinear)

$$g_k(X) + h_k \sum_{i=1}^{n} \left| \frac{\partial g_k}{\partial x_i} \right| \geq 0 \; ; k = 1, .., q$$

Bounds

$x_i - \Delta x_i \leq x_i \leq x_i + \Delta x_i$; $i = 1, \ldots, n$

Minimize: $[\mu_f, \sigma_f]$

Figure 4.9: Formulation for achieving design flexibility [57]

4.3.2 A Modified Robust Design Approach of Achieving Design Flexibility

In this dissertation, a c-DSP robust design formulation is presented. When implemented by optimization, robust design is achieved by bringing the mean on target and minimizing the variance. In the c-DSP formulation, the deviation function is modified to make a tradeoff between various design performances and other goals related with deviation of these performances. The c-DSP formulation can be reformulated as a multi-objective optimization problem shown in Figure 4.10. Compared with original c-DSP, the reformulated c-DSP adds two types of system goals which are the measurements of performance variation and the ratio of performance variation to design variables variation. Correspondingly the deviation function is modified to add weight S_j and the deviation of performance variation Δd^-_j, weight T_j and the deviation of ratio of performance variation ∇d^-_j.

Given

n number of system variables,
q inequality constraints,
m number of system goals,
g(X) system constraint functions,
h_k penalty factors
A(X) system performance measurements
E(A(X) mean value of system performance
Max(A(X)), Min(A(X)) maximum and minimum of performance
G_j goals of system performance
D_j standard deviation of system performance variation
R_j ratio of performance variation to design variable variation
$f(d_j, \Delta d_j)$ function of deviation variables to be minimized

Find

System design variables, x_i i = 1, …, n
System design variable deviation, Δx_i
System goal deviation, d^-_j, d^+_j j = 1, …, m
System goal deviation, Δd^-_j, Δd^+_j j = 1, …, m
System goal deviation, ∇d^-_j, ∇d^+_j j = 1, …, m

Satisfy

System constraints (linear, nonlinear)

$$g_k(X) + h_k \sum_{i=1}^{n} \left| \frac{\partial g_k}{\partial x_i} \right| \geq 0 \text{ ; } k = 1, .., q$$

System goals (linear, nonlinear)

$$\frac{E(A_j(X))}{G_j} + d^-_j - d^+_j = 1; \ j = 1, \ldots, m$$

$$\frac{Max(A_j(X)) - Min(A_j(X))}{2D_j} + \Delta d^-_j - \Delta d^+_j = 1; \ j = 1, \ldots, m$$

$$\frac{Max(A_j(X)) - Min(A_j(X))}{2|\Delta X| R_j} + \nabla d^-_j - \nabla d^+_j = 1; \ j = 1, \ldots, m$$

Bounds

$x_i + \Delta x_i \leq x_i \leq x_i - \Delta x_i$; i = 1, …, n
$d^-_j, d^+_j \geq 0, \ d^-_j \, d^+_j = 0$; j = 1, …, m
$\Delta d^-_j, \Delta d^+_j \geq 0, \ \Delta d^-_j \, \Delta d^+_j = 0$; j = 1, …, m
$\nabla d^-_j, \nabla d^+_j \geq 0, \ \nabla d^-_j \, \nabla d^+_j = 0$; j = 1, …, m

Minimize: deviation function

$$\mathbf{f} = \sum_{j=1}^{m} [W_j(d^-_j + d^+_j) + S_j(\Delta d^-_j + \Delta d^+_j) + T_j(\nabla d^-_j + \nabla d^+_j)]$$

where $\sum_{j=1}^{m} (W_j + S_j + T_j) = 1, \ W_j > 0, \ S_j > 0, \ T_j > 0$

Figure 4.10: Reformulated c-DSP for achieving design flexibility

In the above formulation, G_j and D_j are respectively the mean and the standard deviation of the performance measurement function $A_j(x)$. R_j is the ratio of performance variation to design variable variation. G_j, D_j and R_j are specified by engineering team and works as system goals in the formulation. G_j is the system performance goals that the design solution intends to achieve. D_j and R_j are added system goals that target to an allowable performance variation and a broad design variable variation. In the formulation, the minimization function f is a deviation function that takes all system goals into consideration. To study the variation of constraints, the worst case scenario is considered, which assumes that all variations of system performance may occur simultaneously in the worst possible combination of design variables [58]. To ensure the feasibility of the constraints under the deviations of the design variables, the original constraints are modified by adding the penalty term to each of them where h_k are penalty factors to be determined by the designer. The bounds of design variables are also modified to ensure the feasibility under deviations. Depending on the computation resource, the system goal deviations d_j, Δd_j and $\triangledown d_j$ could be obtained through simulations, analysis or DOE technique such as Response Surface equation. $|\Delta X|$ in the equation is calculated as the magnitude of deviation vector ΔX. The robust design approach introduced in this section is applied to maintain multidisciplinary optimization design freedom to improve the flexibility of a decision making process.

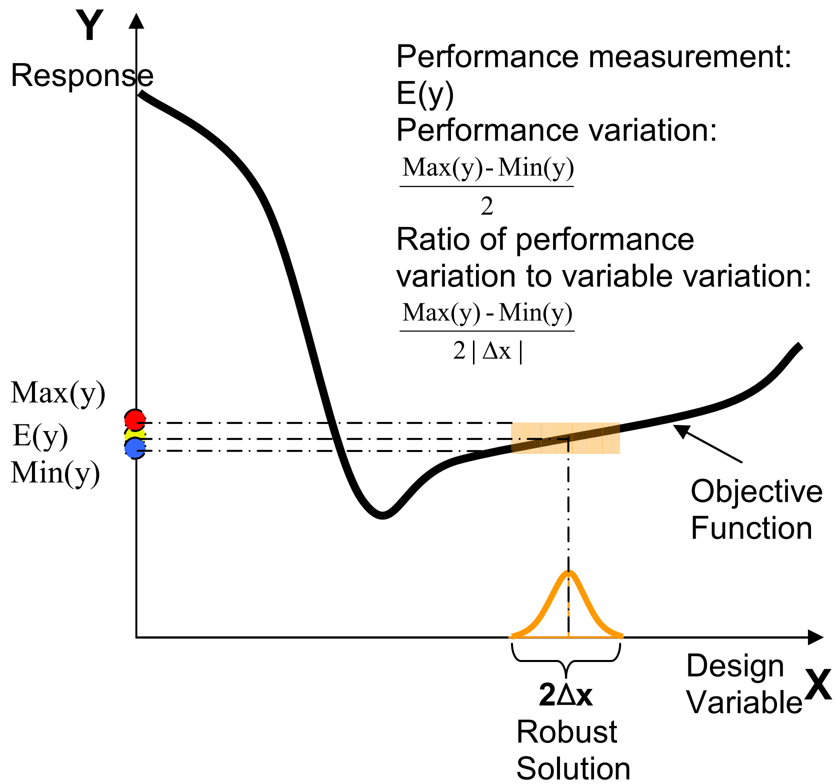


Figure 4.11: System goals in reformulated c-DSP

The c-DSP formulation given in Figure 4.10 is developed to seek the design solution that is robust to the design variation. Illustrated in Figure 4.11, the robust solution is a compromised solution considering three types of system goals. It is close to the global optimal solution so that the performance of robust design satisfies with the engineering requirements. It targets to a specified performance variation so that the design variables can vary within a range. It aims at a specified ratio of performance variation to design variable variation so that large range of design variables is preferable. The robust solution in Figure 4.11 is a tradeoff that considers engineering and robustness requirements of design.

4.3.3 Example of Maintaining Design Freedom in Collaborative Design Decision Making

In this section, the details of how to apply the introduced approach to support engineering teams to make collaborative design decision are given. Assuming leader/follower protocol is selected and the same chip design is the example, this section includes the basic steps of realizing collaborative design decision making, relevant software operation and programming.

4.3.3.1 Response Surface Model and Best Reply Correspondence

Response Surface methodology is "a collection of mathematical and statistical technique that are useful for the modeling and analysis of problems in which a response of interest is influenced by several variables and objective is to optimize this response" [114]. By careful designing experiments and analyzing data, Response Surface methodology allows the relationship between an output variable and many independent input variables to be written in the form of a polynomial function.

$$y = f(x_1, x_2, x_3, \ldots, x_n) = \beta_0 + \sum_i \beta_i x_i + \sum_i \beta_{ii} x_i^2 + \sum_{i,j} \beta_{ij} x_i x_j + \ldots$$

Before design decision can be made, it is necessary for engineering teams to perform design experiments to collect the information of their product design and generate Response Surface model so that design information can be shared by multiple teams. Experiments are designed using Design of Experiments (DOE) techniques. The available choices of experiments include full factorial, fractional factorial, Taguchi Orthogonal Arrays, Central Composite Design, Plackett and Burman (P&B) experiments, etc. The process of constructing a Response Surface model can be divided into several steps.

- Determine the design space and design experiments using DOE techniques;
- Run the computer simulation experiments and gather data;
- Fit the data into Response Surface equation and analyze the significance of regression using ANONA (analysis of variance);
- Run several confirmation tests and develop the final equation.

In this dissertation, the stress design team performs the static analysis and gets the maximum stress. The thermal team performs the thermal analysis and gets highest temperature. By performing the analyses for multiple runs, both engineering teams obtain a set of analysis results. These results are pasted in Table 4.2.

C (mm)	D (mm)	S (MPa)	T (℃)
0.5	0.03	263.612	54.210
0.7	0.03	233.018	57.290
0.9	0.03	251.139	75.279
0.5	0.05	244.778	59.859
0.7	0.05	195.887	63.562
0.9	0.05	185.698	75.218
0.5	0.07	223.258	69.767
0.7	0.07	174.194	67.536
0.9	0.07	158.356	75.274

Table 4.2: Analysis Results

The next step is to fit the data into Response Surface equations. In this dissertation, a statistical software "MiniTab" is selected to generate Response Surface from raw data. Minitab Statistical Software is an ideal package for Six Sigma and other quality improvement projects. From Statistical Process Control to Design of Experiments, it offers the methods to implement every phase of quality project. In addition to more statistical power than some other software, Minitab 14 offers many exciting new features. Among them full function of Design of Experiment is very important in the research of this dissertation.

Before calculating the Response Surface equations, the analysis results are first input into the Minitab. Minitab treats these results as a number of design points and based on these design points Minitab is capable of drawing the surface of solution space. The surface plot of response and input variables are given as the screenshot in appendix of this dissertation.

From the surface plots, it can be found that the stress and thermal responses are not linear to the design variables. Quadratic equations are needed to generate the Response Surface models. Using the function of nonlinear Response Surface regression in Minitab, based on the raw data, Minitab software calculates the following Response Surface regression results.

Response Surface Regression: S versus C, D

```
The analysis was done using coded units.

Estimated Regression Coefficients for S

Term        Coef  SE Coef       T      P
Constant    539.6     74.6   7.228  0.005
C         -653.6    186.5  -3.504  0.039
D        -1424.7   1444.5  -0.986  0.397
C*C        502.7    128.8   3.904  0.030
D*D      21187.9  12876.9   1.645  0.198
C*D      -3276.8    910.5  -3.599  0.037

S = 7.284   R-Sq = 98.6%   R-Sq(adj) = 96.2%
```

Response Surface Regression: T versus C, D

```
The analysis was done using coded units.

Estimated Regression Coefficients for T

Term        Coef  SE Coef       T      P
Constant    62.51    14.12   4.425  0.021
C        -107.94    35.30  -3.058  0.055
D         809.24   273.32   2.961  0.060
C*C       136.80    24.37   5.614  0.011
D*D       865.83  2436.56   0.355  0.746
C*D      -972.62   172.29  -5.645  0.011

S = 1.378   R-Sq = 98.9%   R-Sq(adj) = 97.1%
```

Figure 4.12: Response Surface Regression for Stress and Thermal Analysis Results

According to the calculation results, the Response Surface model for stress and thermal analyses can be written as:

$S(C,D)=539.6-653.6C-1424.7D+502.7C^2+21187.9D^2-3276.8CD$

$=355-113.7C-1599.7D+502.7(C-0.7)^2-3276.8(C-0.7)(D-0.05)+21187.9(D-0.05)^2$

$T(C,D)=62.51-107.94C+809.24D+136.80C^2+865.83D^2-972.62CD$

$=27.4+35C+215D+136.8(C-0.7)^2-972.6(C-0.7)(D-0.05)+866.5(D-0.05)^2$

In a leader/follower design game, the follower is supposed to pass the Best Reply Correspondence (BRC) to the leader. For the thermal disciplines, the system goal is to find the values of variables C and D so that the maximum temperature can be close to 70℃. Within the allowable ranges of variables C and D which are $C\in[0.5, 0.9]$ and $D\in[0.03, 0.07]$, different pairs of values for C and D can be found to make the maximum temperature close to 70℃. If the value of variable D is selected, the value of variable C is obtained by solving Response Surface equation $T(C,D)=62.51-107.94C+809.24D+136.80C^2+865.83D^2-972.62CD$. The value pairs are listed below.

$D=0.03 \rightarrow C=0.5012$

$D=0.04 \rightarrow C=0.5367$

$D=0.05 \rightarrow C=0.5723$

$D=0.06 \rightarrow C=0.6078$

$D=0.07 \rightarrow C=0.6434$

$D=0.03 \rightarrow C=0.5012$

Using linear regression function in Minitab, the above data can be interpreted as linear regression equation C=0.395+3.56D. The regression analysis result is given in Figure 4.13.

Regression Analysis: C versus D

```
The regression equation is
C = 0.395 + 3.56 D


Predictor     Coef   SE Coef       T     P
Constant   0.394530  0.000052  7592.73  0.000
D          3.55500   0.00100  3555.00  0.000


S = 0.0000316228   R-Sq = 100.0%   R-Sq(adj) = 100.0%
```

Figure 4.13: Regression Result of Thermal Analysis

4.3.3.2 Game Construct

After each engineering team performs its engineering analyses, collect analysis data and generate Response Surface model or Best Reaction Correspondence, the design activities of two engineering teams stress and thermal need to be organized to construct design game. In this dissertation, stress team is assumed to play the role of leader and thermal team plays the role of follower. Different with the implementation of leader/follower protocol in the section 4.2.1.3, in this section leader is asked to deliver a range of design solutions to the follower to let follower has the design freedom to make his/her design decision.

The two design variables are assigned to the two engineering teams stress and thermal teams. In this example, design variable C is assigned to stress team and D is assigned to thermal team. Stress team is responsible for selecting a value for the variable C and passes the value of design variable C and an acceptable range of design variable D to the thermal team and thermal team is responsible for selecting a value for design variable D within the range specified by stress team. The design solution template of stress team is illustrated in Figure 4.5 in section 4.1.2.

The design decisions of two engineering teams are constructed into a leader/follower design game where the leader sends the rules which are the value of variable C and range of variable D to the follower and follower sends its BRC to the leader (See Figure 4.14).

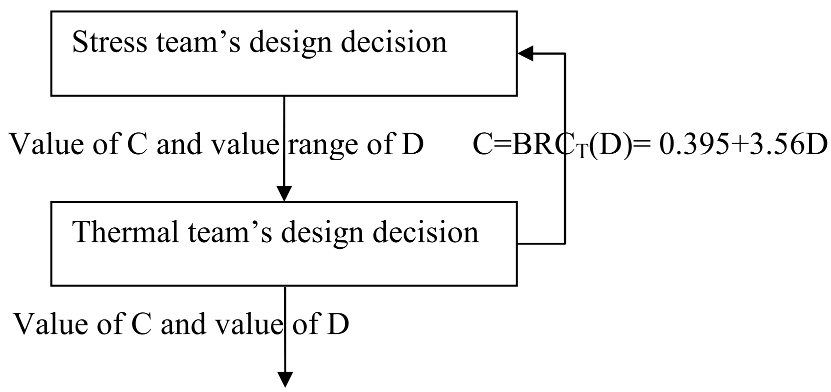


Figure 4.14: Information Exchange in Leader/follower Design Game

4.3.3.3 Optimization Programming

To solve c-DSP formulation, optimization software is needed to find the design solution that satisfies with the constraints and system goals. In this dissertation, the selected optimization software is “VisualDoc” which can perform linear, non-linear, constrained and unconstrained as well as integer, discrete and mixed optimization. Gradient-based, non-gradient based and response surface approximate optimization algorithms are available. In addition a design of experiments module and probabilistic analysis and design capabilities are included.

According to the design game construct, stress engineering team receives the BRC from thermal team and solves the optimization formulation to find an acceptable value for design variable C and an acceptable range for design variable D. The c-DSP formulation of the stress team is given in Figure 4.15.

Given
Package design and material
Find
System design variables, C, D
System design variable deviation, ΔD
System goal deviation, d^-_s, d^+_s
System goal deviation, Δd^-_s, Δd^+_s
System goal deviation, ∇d^-_s, ∇d^+_s
Satisfy
System constraints
$C=0.395+3.56D$
System goals
$S(C, D)/190+d_s^- - d_s^+=1$

$$\frac{Max(S(C,D)) - Min(S(C,D))}{2\times 10} + \Delta d^-_s - \Delta d^+_s = 1;\ j = 1, \ldots, m$$

$$\frac{Max(S(C,D)) - Min(S(C,D))}{2\,|\,\Delta D\,|\times 500} + \nabla d^-_s - \nabla d^+_s = 1;\ j = 1, \ldots, m$$

Bounds
$0.5 \le C \le 0.9$
$0.03 + \Delta D \le D \le 0.07 - \Delta D$
$d^-_s, d^+_s \ge 0,\ d^-_s\, d^+_s = 0$
$\Delta d^-_s, \Delta d^+_s \ge 0,\ \Delta d^-_s\, \Delta d^+_s = 0$
$\nabla d^-_s, \nabla d^+_s \ge 0,\ \nabla d^-_s\, \nabla d^+_s = 0$
Minimize: deviation function
$\mathbf{f} = 0.7(d^-_s+d^+_s)+0.25(\Delta d^-_s+\Delta d^+_s)+0.05(\nabla d^-_s+\nabla d^+_s)$

Figure 4.15: c-DSP Formulation of Stress Discipline

To solve the c-DSP of stress team, the design variables, constraints, system goals are imported into the software VisualDoc (See appendix). The maximum and minimum value of the response is calculated by a DLL program developed by C computer language. The source code of this DLL program is provided in the appendix of this dissertation.

In VisualDoc, the system goal for maximum stress is set as 190MPa. The system goal for deviation of stress is set as 10MPa and the system goal for ratio of stress deviation to design variable deviation is set as 500. The weights in the overall minimized function are respectively 0.7, 0.25 and 0.05. After running VisualDoc analysis, the result is reported in the log file which is given in the appendix of this dissertation (vdoc_26_record). Part of the log file related with the selected value of design variables is shown in Figure 4.16. In the figure, it can be found the value for design variable D is calculated as 0.06623±0.01460. Because the design variable D is constrained within the range of [0.03, 0.07], the acceptable range of design variable D in stress discipline is selected as [0.05164, 0.07]. Correspondingly the value of design variable C is decided by the BRC of thermal discipline which is C equals to 0.395+3.56D. The selected value of design variable C in stress discipline is 0.6308.

```
Independent Design Variables
  1)      0.066233676        0.014596535
Independent Responses
  1)    0.0043000788       0.0038680524         0.3648882
Synthetic Responses
  1)      0.022221478
Combined objective = 0.0222215
```

Figure 4.16: VisualDoc Optimization Results

After the leader stress team runs the optimization software and makes its design decision, the follower thermal design team starts the design works. The thermal team is supposed to follow the rule created by the leader which in our case is $D \in [0.05164, 0.07]$. Through running the optimization problem of thermal discipline, thermal team makes the design decision and selects the value for the design variable D.

Given
Package design and material
Find
System design variables**,** D
System goal deviation, d^-_s, d^+_s
Satisfy
System Constraints
C=0.6307918866
$D \in [0.051637141, 0.07]$
System goals
$T(C, D)/70+d_s^- - d_s^+ = 1$
Bounds
$d^-_s, d^+_s \geq 0, d^-_s\, d^+_s = 0$
Minimize: deviation function
f = $[d^-_s, d^+_s]$

Figure 4.17: c-DSP Formulation of Thermal Team

Using VisualDoc thermal team runs the optimization formulation in Figure 4.17. The result of optimization is D=0.07984. The variable D has a side constraint from 0.05 to 0.07 so that the final value of design variable D can only be set as 0.07.

In this application example, the design solutions for stress team is C=0.6308 and $D \in [0.05164, 0.07]$. The design solution for thermal team is D=0.07. The values of variable C and D are selected as 0.6308 and 0.07 respectively. Detailed steps of applying Response Surface model, design game construct, c-DSP technique, design freedom maintenance approach on collaborative mechanical design are illustrated. It is shown in the example that design decisions can be made separately without requiring to integrate all dependent design activities.

4.4 Summary

Chapter 4 presents the approaches of making design decisions in a distributed design environment. Three introduced collaboration strategies are presented and applied to classify different types of collaborations in design. The approach of generating design freedom for dependent engineering activities is also provided to deliver enough design freedom for accomplishing followers' design activities.

CHAPTER 5
ORGANIZATION OF COLLABORATIVE DECISION MAKING TO FORM DESIGN PROCESS

In the previous chapter, the systematic approach that can support two engineering teams to separately make their design decisions and solve coupled design activities is discussed. From this chapter, the research focus is meta-level mechanical design, the modeling and organization of distributed design activities with complex dependency relationship. The design process is modeled based on the relationship of an important type of design activity, decision making. Other design activities such as information collection, geometric modeling, engineering analysis, and so on are classified into design operations which are usually relevant to certain design decisions and performed for preparing information before decision making or generating product data after decisions have been made. The research in this chapter answers the question (iii) in section 1.3. One assumption made in this chapter is that any design collaboration strategies between engineering teams can be recognized as one of the game theory protocols. The strategy of Concurrent Engineering can be treated as a Pareto protocol and traditionally trial and error strategy can be considered as a special leader/follower protocol without a clear leader's rule.

In this dissertation, Petri-net is chosen as a foundation to develop the design process modeling approach. The Petri-net model that is used to describe design process is called in this dissertation Model of Distributed Design (MDD). In section 5.1.1, a detailed definition of each element in MDD is given. This definition clarifies in a distributed mechanical design how the relationship of design decisions is represented by each Petri-net element. With the MDD definition, design processes are described as a Petri-net models or graphs. These Petri-net models or graphs created based on MDD definition is called MDD or MDD graph in this dissertation. In section 5.1.2, a brief introduction of MDD graph is presented. Each MDD graph is a graphic representation of a design process. The symbols used in MDD graph are explained in details in the section. From section 5.1.3, basic implementations of MDD are illustrated. The presented design process modeling approach is used to describe the application of three design collaboration strategies in design. The MDD graphs based on each type of game protocol or design collaboration strategy are provided as constituent components for describing more complex design processes and show the validation of using MDD to model mechanical design process. From section 5.2, the approach to generate MDD graph alternatives is discussed. For a certain product development, it is likely to generate multiple MDD graphs which represent multiple organizations of all design activities. Each graph is a possible way to organize the design process to accomplish the product development. In section 5.3, the MDD graph alternatives generated in section 5.2 are evaluated. The evaluation criterion are established on the developed measurements of

MDD graphs and the evaluation results are applied to select proper a design process that is effective for a product development.

5.1 Model of Distributed Mechanical Design Process

Following the philosophy of game based mechanical design, the design collaboration strategies between engineering teams are recognized as one of the game theory protocols. Concurrent design is a special Pareto game; trial and error design is a special leader/follower game. Any design collaboration strategies are treated as a type of game protocols.

The design process modeling approach presented in this dissertation is designed to represent the design activity information and design process information. The research goal is to explicitly describe design collaboration strategies that are applied in a distributed design process. As mentioned earlier, Petri-net has been chosen for process modeling because it has the unique advantage of supporting process specification, representation, and evaluation at the same time [115].

5.1.1 MDD Definitions of Place, Transition and Token

In a Petri-net model, place and transition are the two basic types of nodes that need to be defined. In this dissertation, place is defined as design activity, which often requires trade-offs and are represented by the c-DSP formulation. Transition is defined as the design decisions. When a transition is fired, it is expected that certain tokens are transferred from one place to another. With tokens transferred among the Petri-net nodes, the state of Petri-net keeps on changing. Token in MDD is defined as the package of design information that can be used to represent design solution. In order to make a successful collaborative design decision, an important point is to thoroughly understand the partners' design information. In MDD, a transfer of token refers to transfer of values of design variables and some other design information to the dependent design activities. Two types of design variables are defined: master and slave. All values of variables are wrapped in a package and transferred as a token between places. There are two types of variables that are defined in this dissertation. Master variables are those variables engineers in current design activity have privilege of selecting their values. Slave variables are those variables their values are controlled by engineers in some other design activities and cannot be selected by engineers in current design activity. The decision maker in current design activity neither has the privilege to modify nor select the decided values of the slave variables. However, engineers in current design activity can specify the acceptable range or options for slave variables. In leader/follower strategy, slave variables are used to transfer design rules from leader to follower. For these slave variables, the leader can specifies the feasible solution sets or range for the follower and let follower decide their values. In this situation, in leader's design activity, these variables are slave variable and in follower' design activity, these variables are master variables.

The place and transition are connected by arcs which can be weighted. In MDD, the weight of an arc, which connects a design activity to a decision, is defined as the number of design solutions that are required to start a consecutive decision making. The weight of arc which connects a design decision to a design problem is defined as the number of design solutions that are generated by a decision. The default weight for each arc is 1. In addition, two properties are defined in MDD: the assigned designers for a design problem and the kick off and elapsed time of a decision. With these definitions of place, transition and token in a collaborative decision making process, the MDD is mathematically defined as a 7-tuple, $\mathrm{MDD} = (\mathrm{P, D, F, U, E, K, M_0})$, where

$\mathrm{P} = \{p_1, p_2, \ldots, p_m\}$ is a finite set of design activities.

$\mathrm{D} = \{d_1, d_2, \ldots, d_n\}$ is a finite set of design decisions.

$\mathrm{F} \subseteq (\mathrm{P} \times \mathrm{D}) \cup (\mathrm{D} \times \mathrm{P})$ is a set of arcs, which denote token flows.

$\mathrm{U} = \{u_1, u_2, \ldots, u_r\}$ is a set of engineers who are responsible for solving the design activities.

$\mathrm{E} : \mathrm{D} \rightarrow \{tp_1, tp_2, tp_3, \ldots\}$ are elapsed time attached to design decisions in D.

$\mathrm{K} : \mathrm{D} \rightarrow \{t_1, t_2, t_3, \ldots\}$ are kick off times attached to design decisions in D.

$\mathrm{M_0} : \mathrm{P} \rightarrow \{0, 1, 2, 3, \ldots\}$ are the initial markings, the number of tokens in each place, which are design activities in P, at the beginning. The MDD with the given initial marking is denoted by $(\mathrm{MDD}, \mathrm{M_0})$.

Token and design activity are formulated using solution template and design activity template introduced in Chapter 4. The solution template is 3-tuple including the values of master variables, a set of values for slave variables or a value range for slave variables and the necessary design decision information about Response Surface or Rational

Reaction Set, $T = (Mv, Sv, RRS/RS)$. In the template T, Mv represents the values of master variables. Sv represents a set of values for slave variables or a value range for slave variables. The third part of token definition can be RRS or RS depends on different collaboration strategies are used. RRS represents the Rational Reaction Set and RS represents the Response Surface of objective function in a design activity.

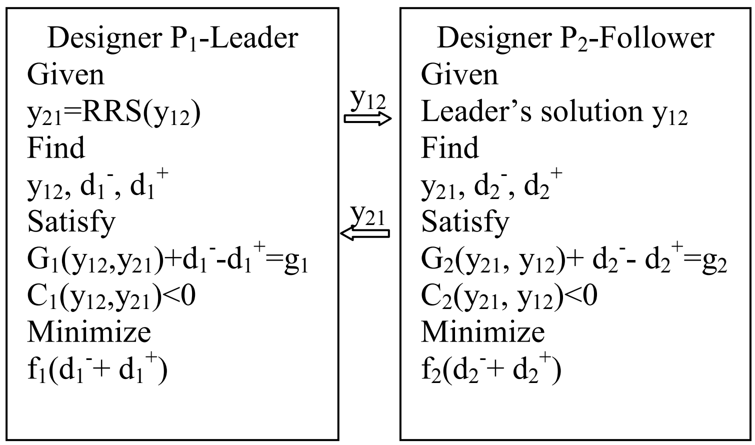


Figure 5.1: Compromise DSP in Leader/follower Collaborative Decision

Using Figure 5.1 as an example, y_{12} and y_{21} are system variables. C_1 and C_2 are constraints including bound constraints for the system variables. f_1 and f_2 are functions that are to be minimized. The leader's formulations and the follower's are slightly different. The leader's formulation has the Rational Reaction Set inputted from the follower. The follower's formulation uses leader's solution rang as given condition. In this example, the token transferred from activity P_2 to P_1 is the Rational Reaction Set $y_{21}=RRS(y_{12})$ and the token transferred from activity P_1 to P_2 is the design rule y_{12} which are the values for a set of master variables in P_1.

5.1.2 MDD Graph

Based on above definitions in section 5.1.1, it is possible to describe a design process using a Petri-net graph. Illustrated in the Petri-net shown in Figure 5.2, the design activity P_1 is represented as a circle. The symbol "|" represents design decisions. In the figure, the design decision is D_1. For each design activity, design solutions are assigned to it and specified as tokens. In Figure 5.2, T_1 is a token related to P_1. n_1 is the number design solutions that can fire the design decision D_1. n_2 is the number of design solutions that are generated by decision D_1. The default value for n_1 and n_2 is 1 if they are not explicitly specified. By choosing different values for n1, the condition of firing decision D_1 is changed and it is possible to delay activity P_1 till all design information is ready. By choosing different values for n_2, it is possible to let activity P_1 to generate multiple solutions so that in consecutive activities engineers can have more choices. The solid dot in the design activity P_1 indicates that there is one token in other word design solution in activity P_1 at the initial state. For each design decision such as D_1, a kick off time t_1 and elapsed period tp_1 can be assigned as a property of a decision. In order to describe who is responsible for a design activity, a property $\{u_1\}$ is attached to P_1 to represent engineer u_1 needs to work on the activity P_1.

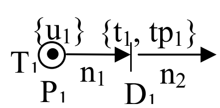


Figure 5.2: Simple Decision Making Model

In some Petri-net applications, the connections of place and transition do not have any restrictions. One place can connect to multiple transitions and one transition can connect to multiple places. However in a design situation, one design activity connects with multiple design decisions will cause the conflict, which means selecting multiple values for one design variable. To avoid the conflict, the restrictions on MDD graphs are required. In this dissertation, the restrictions are made as: except for some special connecting activities in MDD which will be introduced in section 5.1.3, an engineering design activity has only one design decision attached to it, while a design decision can pass tokens or solutions to multiple consecutive design activities. The reason to make this restriction is that multiple decisions attached to one design activity will cause conflictions in a design decision making process. The tokens that are initially placed in various design activities are blank design solutions. These design solutions do not specify any values for design variables.

Using Petri-net graph symbols, each design process is first defined as a Model of Distributed Design (MDD) and then each MDD is translated into a MDD graph which contains basic design information and reveals the connection relationship of multiple design activities. In mechanical design, MDD graphs provide engineers with clear and detailed information about how design process is actually implemented. Its function is like the function of a map helpful for travelers to find right path. Using MDD graph, engineers are able to describe an existing design process and identify the bottleneck of it. Based on accurate information further improvements can be made and exiting design process can be modified. For a new product development, using MDD, engineers can generate multiple design process alternatives so that suitable design process can be selected. In this dissertation, MDD graphs of three types of design collaboration strategies are provided in section 5.1.3 which is an example of using MDD graphs to describe existing design process. The MDD graphs of three types of design collaboration strategies are treated as constitutional elements to describe more complex design process and different combination of three strategies in design process can generate various design process alternatives which are helpful for process selection.

5.1.3 MDD Graphs of Design Games

Figure 5.2 illustrates a single design activity in which no engineers from different design disciplines interacting with the responsible engineer in the decision making process. In this section, consider the situation of two engineers simultaneously working on different design activities that are coupled and pursuing the optimal results through decision interactions. There are many ways to organize or construct separate decisions into a collaborative game. The ideal case is the full cooperation in which all engineers share design information and design variable access so that their design activity can be combined. However, this full cooperation is restricted by the increase in computing expense and organization difficulties [116].

In this dissertation, design collaboration is achieved through setting up proper strategies which include three types of game protocols: Pareto cooperative, Nash noncooperative and Stackelberg leader/follower. In the rest of this section, MDD graphs are used to model the decision making process based on three game protocols. The three types of design collaboration strategies are building elements for MDD to construct complex collaborative design processes.

Pareto Decision Making

The MDD graph of a Pareto strategy is illustrated in Figure 5.3. The two activities P_1 and P_2 are coupled. Engineers u_1 and u_2 make design decisions to update the design variables till the Pareto condition are achieved. Once a design solution is found, a connecting activity is included to check whether constraints and performance are satisfied for both activities P_1 and P_2. This connecting activity is designed to have two alternative decisions. If satisfaction decision is made, the design process moves to the next stage. If not, the design moves to the complimentary design activity - the design variables and Response Surface of the minimize function in one design activity are transferred to the other for further design improvement. Thus in Pareto design strategy, design activities are integrated and solved together.

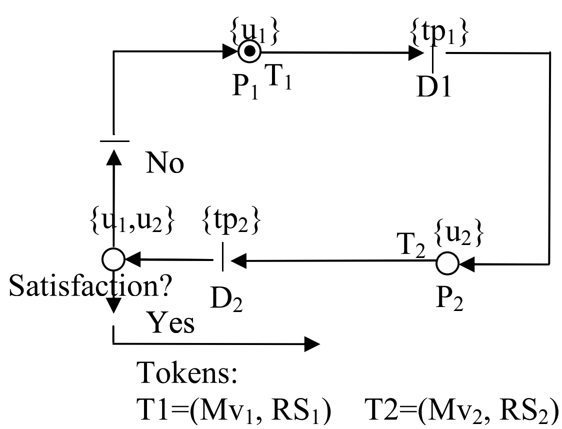


Figure 5.3: Pareto Decision Making Model

In this game construct, one engineer first passes design information to the other and then two design activities are combined and solved together. Pareto design strategy needs to transfer the design Response Surface information from one activity to the other. The information exchange is the key to obtain quality product design. In a traditional design environment, information exchange is achieved by face to face meeting. In a distributed design environment, Pareto design strategy may need extra time on information exchange on internet. Therefore the design efficiency of Pareto strategy is restricted in some cases.

Nash Decision Making

The MDD graph of the Nash design strategy is illustrated in Figure 5.4. Two coupled activities P_1 and P_2 need to be solved by two independent engineers, u_1 and u_2. Through decision D_1 engineer u_1 generates n_1 tokens or in other words n_1 design solution points to form a solution space. Correspondingly u_2 generate n_2 tokens for its solution space. All these solution points are represented using design solution template and passed between design activities as tokens. A special intersection activity is defined to find the intersection of two solution spaces which satisfies with the Nash condition. This special design activity has two decisions connected with it. Respectively these two decisions represent yes and no responses to Nash condition. The design decisions in Nash stategy can be made simultaneously which means D_1 and D_2 in Figure 5.4 may have the same kickoff time.

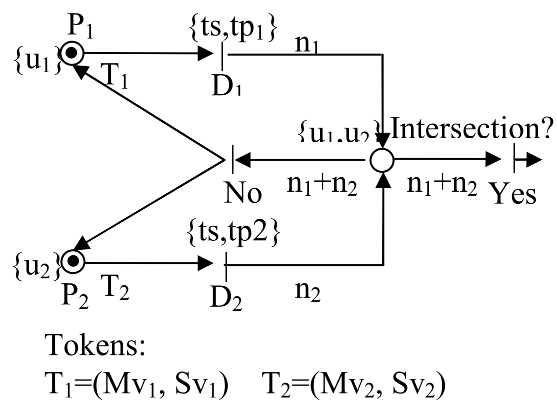


Figure 5.4: Nash Decision Making Model

Stackelberg Decision Making

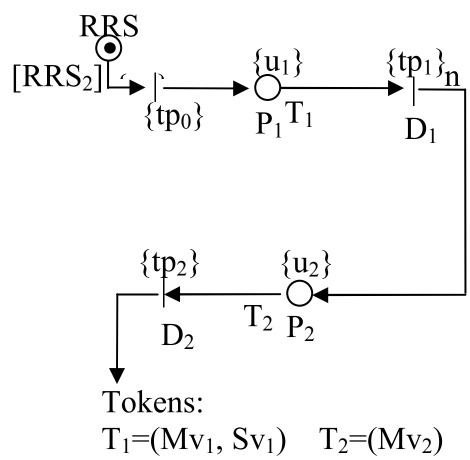


Figure 5.5: Stackelberg Decision Making Model

The Stackelberg design strategy is natively a sequential process. The MDD graph of the Stackelberg design strategy is illustrated in Figure 5.5. Engineer u_1 plays the role of a leader and sets a rule for the follower u_2. The leader's rule includes n tokens, which cover n preferable solution points in activity P_1. The follower u_2 receives the rule from the leader u_1 which is the preferable solution points of the leader team represented by solution template. Then the follower solves P_2 to find the best design within the preferable solution space given by u_1. In Figure 5.5, before the leader generates design rules, the follower first generates the Rational Reaction Set in his/her discipline and transfer this information to the leader to help leader make accurate decision.

The detailed information exchange in Stackelberg strategy can be found in Chapter 4 which includes more discussion about the information that are needed to support collaborative design decision making.

5.2 Approach of Generating Alternative MDD Graphs

In section 5.1.3, MDD graphs are used to model three basic design constructs between two engineering designers or teams. From section 5.13, it is obvious that MDD graphs can represent simple design processes, describing accurate design information and dependency relationship between activities.

Besides representing existing design process, a further research is to develop an approach to generate different MDD graphs based on the information of design activities which can be collected in the detailed design stage. These MDD graphs are considered as process candidates for selecting a proper design process for certain product development. In this dissertation, MDD graphs do not contain design iterations. From our understanding of a proper design process, the generated design processes are not intentionally created to have iterations. In practice, it is possible that engineers find that it is necessary to redo some part of design works. However, in these cases the iterations are ad-hoc, not planned before the implementation of product design. One benefit of applying the collaborative design decision making approach which is introduced in Chapter 4 is that by doing so product design may avoid unnecessary iterations. The approach in this section provides a possible way to generate multiple MDD graphs without design iterations.

The rest of this section is organized as follows. In section 5.2.1, analyzing the dependency relationship for design activities is discussed. To clearly explain the analysis approach, a simple example with six design activities and eight design variables is presented. As starting point of generating multiple MDD graphs, the dependency relationship analysis is the key for finding different combinations of design activities in a design process. In section 5.2.2, generating MDD graphs based on the dependency relationship analysis result is provided. The generated MDD graphs in section 5.2.2 are many options for engineers to implement a product design. These options are not restricted by any practical constraints such as resource limits, schedule confliction and so on. By applying some constraints in section 5.2.3, the MDD graphs that can satisfy with the requirements are selected from the generated MDD graphs.

5.2.1 Analysis of Dependency Relationship

In DBD, the major activity of a mechanical design is design decision. Each design decision can set the values for a certain group of design variables and thus provide a solution for answering a product design generally. In this dissertation, the design activity is explicitly defined in the design activity template and the result of design activity is explicitly defined as solution template. A design process is an organization of the design activities and a flow of design information exchange between design activities.

When a design process is considered as an organization of design activities represented by activity and solution templates, an important fact is the dependency relationship of all design activities largely influences the organization of activities. Because of the sharing of design variables, engineering design activities are usually dependent with each other and collaborative decision making is required to perform the coupled design activities. Finding the dependency relationship of all design activities based on the design variables that are used is important to generate feasible design processes.

To analyze the dependency relationship, the information about the engineering design variables used in each design activity is required to be collected. This design variable information is available in detailed design stage because in this stage design activities are formulated using c-DSP templates and c-DSP templates contain the design variable information as well as some other information such as design goals, design constraints are all defined in the c-DSP activity template.

Because the dependency is caused by the sharing of design variables, listing all design activities with the corresponding design variables provides useful information. In this dissertation, the relationship information of activity and variable is given in a table like

Table 5.1. Table 5.1 is a description of the activity and variable relationship, which is provided as an example in this chapter.

Variables	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
P_1	x		x		x			
P_2		x		x				
P_3	x					x	x	
P_4		x		x		x		x
P_5	x		x		x		x	
P_6		x		x				x

Table 5.1: Activity and variable relationship

From the example, design activity P_1 uses variables v_1, v_3, v_5; design activity P_2 uses v_2, v_4; design activity P_3 uses v_1, v_6, v_7; design activity P_4 uses v_2, v_4, v_6, v_8; design activity P_5 uses v_1, v_3, v_5, v_7; design activity P_6 uses v_2, v_4, v_8. If two design activities have shared variables, these two design activities are coupled or dependent with each other. Decision collaboration strategies are needed to solve coupled design activities. One type of design collaboration strategy is assigned for each pair of design activities which have shared variables.

However, if design collaboration strategies are selected for each pair of coupled design activities, the whole design process becomes a redundant process because one design variable can be shared by more than two design activities and the value of this design variable is selected in the design collaboration strategies of multiple pair of coupled design activities. To simplify the design process and make the design process more efficient, a possible consideration is to construct design process to include the design activities with the same shared variables into one design collaboration strategy. These bigger design constructs combine the interactions of each pair of coupled design activities and simplify the design process. Besides gathering the dependent design activities, application of Stackelberg strategy is the other possible manner to avoid unnecessary redundant decisions on the shared design variables. The values of the shared design variables can be first preliminarily selected as an acceptable range or options. These range or options are transferred to the design activities in the following disciplines as design rules. In this way, the final values of any shared design variables are specified only one time and different part of design activities are linked by the design rules.

To apply the above two considerations into MDD graph generation, it is required to know what design activities need to be gathered and solved together and what activities need to be separated and solved in different design collaboration strategies. Different combination of design activities can result in different MDD graphs. If more feasible combinations can be found, more MDD graph alternatives can be generated.

Because combinations of design activities can only be made based on shared design variables, after the activity and variable relationship is given in Table 5.1, it is helpful to clarify the maximum shared design variables for each group of design activities. In this dissertation, the maximum shared design variables are defined as a group of design variables which are shared by a group design activities and no other shared design variables can be found for the same group of design activities. Based on the information provided by Table 5.1, a maximum shared design variable graph which describes the maximum shared design variables between engineering design activities is given in Figure 5.6 in which each node in the graph is a representation of a set of shared design variables that are used in a certain set of design activities. These shared design variables are maximized since it is impossible to find more shared variables between the certain set of design activities. As an example, in Figure 5.6, design variable v_6 is shared by design activities P_3 and P_4. And it is impossible to find any design variables beside variable v_6 that are shared by activities P_3 and P_4. According to the definition, variable v_6 is a maximum shared variable between P_3 and P_4. For the same reason, variables v_1, v_3, v_5 are included in activities P_1 and P_5 and no other shared variables can be found in P_1 and P_5 so v_1, v_3 and v_5 are maximum shared variables.

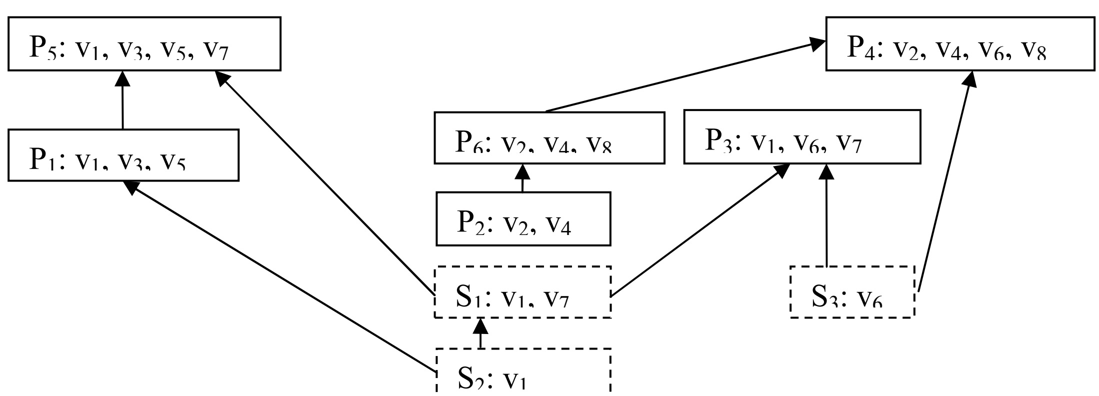


Figure 5.6: Maximum shared variable graph

The maximum shared variable graph in Figure 5.6 describes the dependency relationship between design activities. It is different from Design Structure Matrix (DSM) [95] or other CE complex design process research approaches [96]. It describes more clearly how design activities are related with each other. The dependency relationships of more than three design activities are discovered from the graph. For example, in Figure 5.6, it can be found v_1 is shared by three activities P_1, P_5 and P_3. However DSM can only discover the relationships of a pair of activities. In a DSM graph (see Figure 5.7), the relationships of three design activities are not properly represented.

The detailed explanation of maximum shared variable graph is provided as follows. In Figure 5.6, the seven design activities are represented by six solid rectangles. Each arrow in the graph represents that the design variables in one node is included in the design variables in another node. Besides solid rectangles, there is another type of nodes in the dependency relationship graph which are represented by dotted rectangle. These dotted rectangles are maximum shared design variables that can be found between multiple design activities. Using this dependency relationship graph, it is easy to find the maximum shared variables for any sets of design activities in product development.

	P_1	P_2	P_3	P_4	P_5	P_6
P_1	1	0	1	0	1	0
P_2	0	1	0	1	0	1
P_3	1	0	1	1	1	0
P_4	0	1	1	1	0	1
P_5	1	0	1	0	1	0
P_6	0	1	0	1	0	1

Figure 5.7: Design Structure Matrix

5.2.2 Generation of MDD Graphs

After the dependency relationships are obtained, the next question is how to generate multiple MDD graphs based on the information. From the discussion in section 5.2.1 it can be found to select a value for a share design variable all design activities that use this shared variable need to be involved in. In Figure 5.6, v_1 is shared by three design activities P_1, P_3, P_5. v_1 and v_7 are shared by two activities P_5 and P_3 and v_6 is shared by two activate P_3 and P_4. Because the design variable v_1 is shared by design activities P_1, P_3 and P_5, three design activities P_1, P_3 and P_5 need to coordinate with each other to decide v_1's value. The design activities P_1, P_3 and P_5 are required to be solved together. In this example, starting from node v_1 in Figure 5.6, P_1, P_3 and P_5 can be categorized into the first group of design activities that need to be performed. When P_1, P_3 and P_5 are all solved, the values of v_3, v_5 and v_7 can be also decided in the process of searching for v_1. Only one variable v_6 cannot be fully decided because v_6 is also a design variable shared by design activity P_4 which is not a member of the first group. To find the value for design variable v_6, all design activities that use v_6 need to be categorized into the second group. Because P_4 is the only design activity besides P_3 that uses design variable v_6, only P_4 needs to be solved. In this example, P_4 is categorized into the second group of activities. After P_4 is solved, it is obvious that the values of design variables v_2, v_4 and v_8 cannot be decided in P_4. To select the values of v_2, v_4 and v_8, P_2 and P_6 need to be solved. In this example, P_2 and P_6 are categorized into the third group of activities. Starting from variable v_1, three groups of design activities are identified according the dependency relationship of design activities. Each group of activities are qualified to select the values for a certain set of design variables and design activities in each group are constructed using a design collaboration strategy. The design variables which are shared between activities in different groups are solved using Stackelberg leader/follower design collaboration strategy. A brief description of categorizing design activities into different group starting from finding values for maximum shared design variables is given in Figure 5.8. An illustration of grouping design activities from maximum shared variable v_1 is given in Figure 5.9.

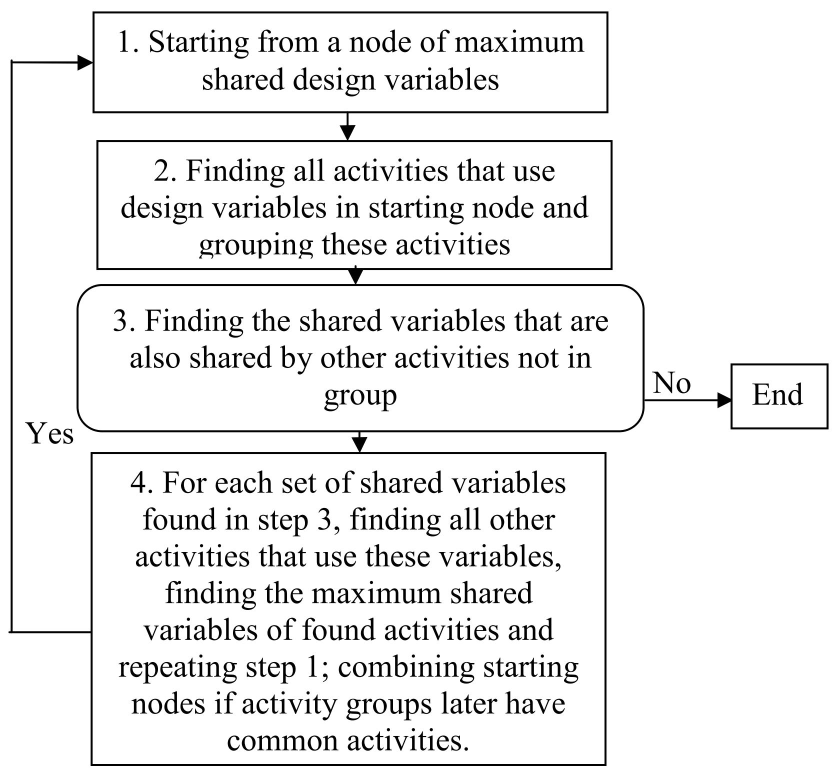


Figure 5.8: Grouping design activities

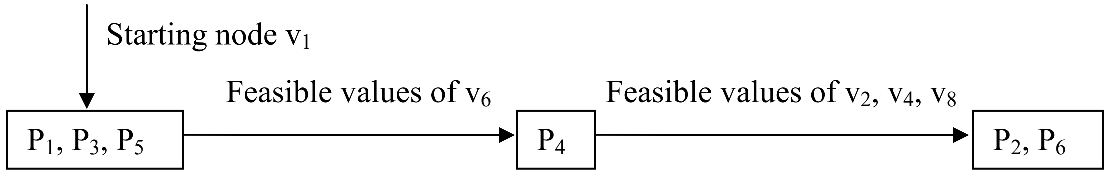


Figure 5.9: Grouping design activities from v_1

By first choosing different node in Figure 5.6, different variable groups can be found using the same procedure defined in Figure 5.8 and through different groupings different MDD graphs can be generated later. For example, if maximum shared design variable v_6 is selected as starting node, the first group of design activities would be P_3 and P_4. There are two second groups of design variables respectively P_1, P_5 and P_2, P_6. Following the step 4 in Figure 5.8 if in these two groups there is not a common design activity, they can be solved separately. The illustrative graph is given in Figure 5.10.

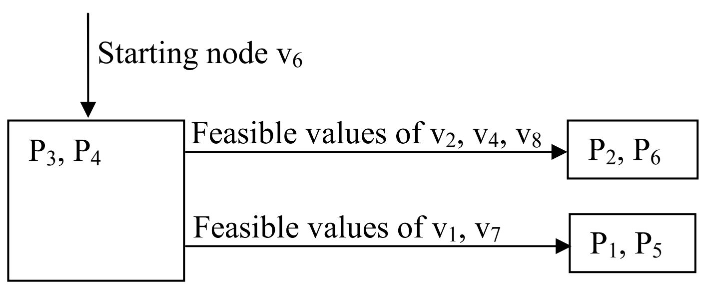


Figure 5.10: Grouping design activities from v_6

If the first set of design variables is selected as v_2 and v_4, the overall design activities can still be divided into three groups. The following figure is the illustration based on this assumption.

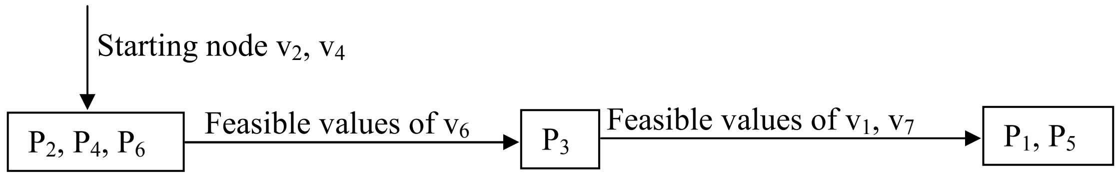


Figure 5.11: Grouping design activities from v_3 and v_4

There are 9 nodes in Figure 5.6. Each of them can be chosen as the starting points to generate design activity groups. In this example, three of them are illustrated and explained above. The rest six of them is given in Figure 5.12. It is important to mention that not all generated activity groups are different. Some of them are the same. Altogether there are 5 different activity groups can be generated from 9 starting nodes.

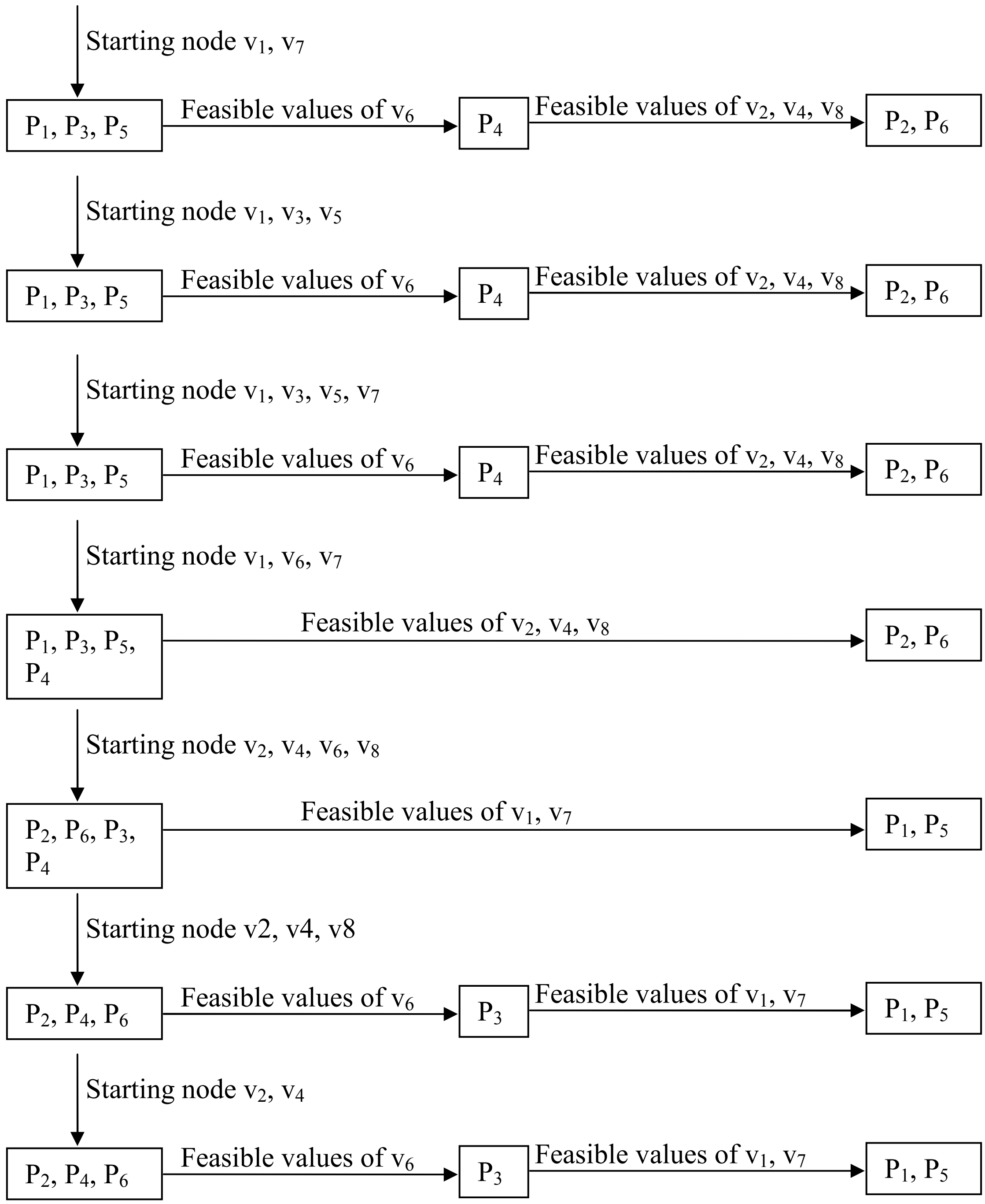


Figure 5.12: Generating activity groups

Each of the above graphs can result in a set of different design processes. The design collaboration strategy between two groups of design activities can be selected as leader/follower because feasible values of design variables are passed from one group of design activities to the other. Within one group of design activities, it is possible to use all three types of design collaboration strategies, Pareto, Nash and leader/follower. If different groupings and strategies are corresponding to different MDD graphs, the number of all MDD graphs that are generated using the approach in this dissertation can be estimated. In the example of this chapter, since there are 5 different groupings and in each grouping there are two or three groups that are organized using three strategies of design constructs, the maximum number of MDD graphs that can be generated is: $3^3+3^3+3^2+3^2+3^3=99$. 99 is the maximum of MDD graphs that can be generated. Among these 99 alternatives, not all alternatives are unique. From the example, it can be found in some groupings it is likely that there is only one design activity in a group. In this case, design collaboration strategies cannot be applied in the group which has only one design activity and number of MDD graphs is reduced. For the groupings in Figure 5.9 and 5.11, both of them have three groups but only two of the groups have multiple activities. Because of this reason the number of MDD graphs is reduced to 63. Generally in MDD graph generation, if number of grouping is “k” and number of groups in grouping “i” is “g_i”. The maximum number of MDD graphs is calculated as $\sum_{i=1}^{k} 3^{g_i}$

Based on the activity groups illustrated in Figure 5.12, if the grouping in Figure 5.9 is selected, one MDD graph alternative that represents a feasible design process is given in Figure 5.13. The generation of MDD graph in Figure 5.13 starts from choosing the strategy for each activity group to initially generate MDD graphs and then detailed information of design process such as definitions of tokens are included to complete the

MDD graph generation. Figure 5.13 is an example of the first step. In this step, Pareto strategy is chosen for activities P_1, P_3 and P_5 and Nash strategy for activities P_2 and P_6. The leader/follower strategy is used to connect different groups of activities. Figure 5.13 is one of the MDD graphs that can be generated using the approach. By choosing different strategies for each group of activities, multiple MDD graphs can be generated based on the same groupings. From Figure 5.13, three coupled design activities P_1, P_2, and P_3 are solved using Pareto strategy. The design information of these three activities is transferred from one to another using solution tokens. After all information is collected in the activity P_5, engineers in P_5 combine the considerations of from all three activities and attempt to find a solution that best satisfies with all requirements. Once the design solution is made by engineers in activities P_1, P_3 and P_5, the solution is passed to activity P_4 in which engineers further select the value for variable v_6. In the final part of Figure 5.13, the design is constructed using Nash strategy. An intersection of the solution spaces needs to be found.

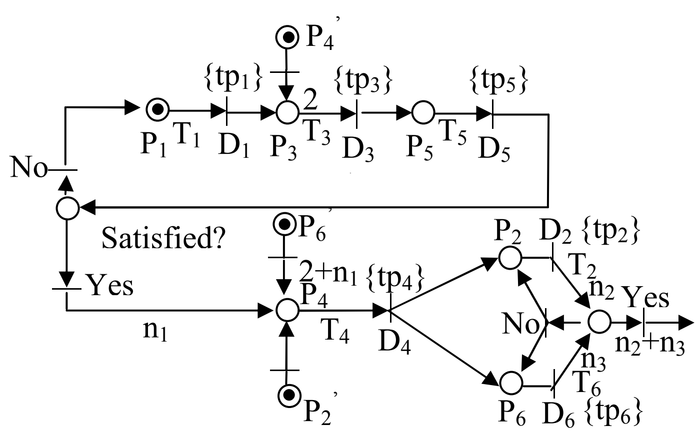
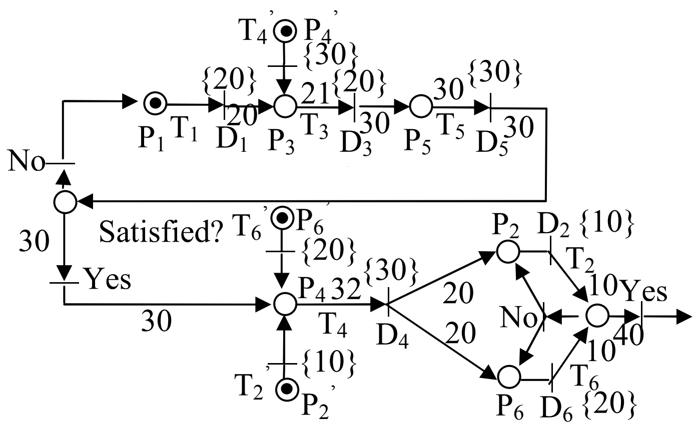


Figure 5.13: A MDD graph candidate

After a MDD graph candidate is generated, detailed information are required to complete the MDD graph generation. One type of detailed information is the master and slave design variables which need to be specified in the tokens of each design activity. As mentioned before, the values of master variables can be decided and the values of slave variables can only be ranged by the engineers in the design activity. The definition of tokens explicitly describes in which activity the values of design variables are selected. Other types of information include the responsible teams or engineers for each activity, elapsed and start time for each design decision, and number of tokens that need to be generated by each design decision. To specify the additional information, it is assumed that engineering experience is important. Existing design process and previous experience are helpful for giving a common sense of how long a design activity lasts and how many solution points are essential for further design activities. A complete MDD graph with all information added is illustrated in Figure 5.14. In Figure 5.14, tokens T_1, T_2, T_3, T_4, T_5 and T_6 are defined. Variables v_1 to v_8 are assigned as master or slave variables in each token. In Pareto strategy, Response Surface of each design activity is passed to increase the understanding of design activities in different disciplines. Because leader/follower strategy is used, followers are asked to pass their Rational Reaction Sets to the leaders. T_4', T_2' and T_6' are those tokens which contains followers' RRSs.



Tokens:
T_1-(Mv: v_1; Sv : v_3,v_5 ; RS: $RS_1(v_1,v_3,v_5)$)
T_3-(Mv: v_7; Sv: v_6; RS: $RS_3(v_1,v_6,v_7)$)
T_5-(Mv: v_3,v_5; RS: $RS_5(v_1,v_3,v_5,v_7)$)
T_4-(Mv: v_2,v_6; Sv: v_4,v_8; RS: $RS_4(v_2,v_4,v_6,v_8)$)
T_2-(Mv: v_4; Sv: v_8; RS: $RS_2(v_2,v_4)$)
T_6-(Mv: v_8; Sv: v_4; RS: $RS_6(v_2,v_4,v_8)$)

T_4'-(RRS: $v_6=RRS_4(v_2,v_4,v_8)$)
T_2'-(RRS: $v_4=RRS_2(v_2)$)
T_6'-(RRS: $v_8=RRS_6(v_2,v_4)$)

Figure 5.14: Complete MDD graph candidate

5.2.3 MDD Graphs Satisfying With Constraints

In practice, product designs have various constraints on the design processes. These constraints can be divided into two types. Some are related with sequence, which means design activities need to be completed in a specific order. Some are related with the capability such as limitations on the number of concurrent design activities, the maximum number of generated design solutions and the shortest elapse time of design activity and so on.

The sequence constraints on the MDD graphs can be applied in the process of MDD generation. After the groupings of design activities are obtained, the sequence of performing design activities is also decided. The activities in the leading groups are always performed early than the activities in the following groups. By keeping the groupings which satisfies with the sequence condition and removing the rest, engineers generate the MDD graphs which exactly conform with the sequence constraints.

The capability constraints are applied after all MDD graph candidates have been generated and proper evaluations have been done on these graphs. These evaluations which are discussed in the following section are requested to provide the various performances of each candidate to look for a MDD graph that satisfies with all capability constraints and targets to better performances.

5.3 MDD Evaluation and Analysis

After MDD graphs have been generated, the selection of suitable MDD graph is based on the performances evaluations. Modeling of an existing design process helps engineers better understand many design process characteristics so that further improvement can be made. Generating MDD graphs provides design process alternatives so that further selection for better performance graph can be made. A major strength of applying MDD graphs is that MDD graphs are Petri-net models and various analyses are available for Petri-net to identify and evaluate performances. From the survey of Petri-net research, there are different types of analysis or properties. In general, two types of properties are studied with a Petri-net model: properties that depend on the initial marking and properties that are independent of the initial marking [66]. Most of the properties presented in this section belong to the first type, initial marking dependant. This is because product development in practice has a clear initial stage and later a complete stage. It cannot be modeled as some state machines which only repeat the transformations from one state to the other. Design practice is a key consideration when MDD graphs are applied for describing and analyzing a design process. Any applications of MDD graphs are related with practical design processes. Following this principle, four properties are defined as measurements to evaluate MDD graphs. These properties are introduced in the sections 5.3.1 to 5.3.4. Currently many Petri-net tools are available for commercial or research purposes, the defined prosperities are able to be calculated by these powerful tools.

5.3.1 Evaluation 1: Quality of Design Solutions

Product developments have many quality standards that need to be meet with. A product design which takes about one day is hard to have the same quality as a design which takes about one month. There should be a measurement about the quality of design solutions.

There are many factors that can influence the quality of product design. A major type of factors is related with human. Their experience, skills, knowledge, capabilities are all foundation to achieve quality of design. Their effective communication and coordination are important to avoid design failures. The human related factors influence the quality of any product design regardless of which design process is used in product development. In this section, the focus is the influence of different design processes on the quality of design. For the same group of engineers, it is possible to implement different product design processes and the quality of product design varies with the selection of design process. In section 5.2, the approach that can help generate MDD graph candidates is provided. Each MDD graph candidate is a representation of a potential design process. By implementing different design processes, it is expected different product designs will outcome.

There are some factors that are related with design process and can influence the quality of product design. Design freedom is one of factors that are related with design process. Engineers performing the design activities in the early design stage have more freedom to choose their preferable solutions. As product development goes on, engineers performing the activities in later design stage will find the design works are fixed, lack of the freedom to make changes or modifications because of the high cost of reworks. Compromising design qualities is the only way to release the product design in time.

Estimating the quality of design solution before product development actually starts is difficult. Engineers' design decisions are human behaviors which can be influenced by the design knowledge, experience and personal preference and hard to estimate before design starts. However the design freedom of each design activity is mainly decided by the organization of design process and thus it is possible to estimate the design freedom before design starts.

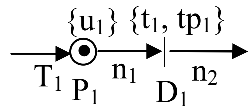


Figure 5.15: A MDD graph of decision making

In Figure 5.15, the information of design activity P_1 and design decision D_1 are provided which include the number of input tokens n_1, number of output tokens n_2 and total number of coupled design variables in MDD graph. Based on the number of input and output tokens, there exists an approximate relation between design freedom and solution tokens. It is found that input tokens have their positive influence on design freedom and more input tokens will increase the design freedom; similarly output tokens have their negative influence on design freedom and more output tokens will decrease the design freedom. Illustrated in Figure 5.15, as n_1 increases, the engineers in activity P_1 have more design space to find a suitable solution for design and thus the design flexibility for activity P_1 increases. It is expected a better solution can be obtained with increasing n_1. As n_2 increases, engineers in activity P_1 need to generate more solution tokens for delivering design freedom and more solution tokens compromises the performance of product design in activity P_1. It is expected that a quality loss of design in activity P_1 if engineers need to generate more tokens which deviate from the optimal solution.

As to the influence of the number of design variables on design freedom, it is also possible to find a relation between design variables and design freedom. Design variables are assigned to each design activity as master variables or slave variables defined in the solution tokens. For master variables, engineers can select their final values and thus they have more freedom to obtain better design solution. When number of master design variables in activity P_1 increases, it is expected that a better solution can be generated. For slave variables, engineers can only specify a range or options for their values and thus they don't have full freedom to search for quality product design. In this dissertation, if the design freedom of one master design variable is treated as 1, the design freedom of one slave variable is C_f which means on average one slave variable only brings C_f part of the design freedom for a design activity. In this dissertation, C_f is set as 0.5. The number of master and slave variables can influence the design freedom which is an important factor for design quality. More slave design variables in a design activity cause the loss of product performance while more master design variables provides more possibilities for optimal design solution searching.

Based on the discussion of the various relations between design activity information and design quality, an estimation of the design freedom in each design activity is given as the following equation. This equation cannot exactly describe the design freedom in each design activity. It approximately evaluates the design freedom.

$$S_i = \frac{n_{input}(n_{master} + C_f n_{slave})}{(n_{input} + n_{output})(number\ of\ \text{var}aibles\ in\ activity\ i)}, n_{slave} > 0, n_{master} > 0, n_{input} > 0, n_{output} > 0$$

The quality of design in activity P_i is measured using the design freedom S_i which is calculated in above equation. n_{input}, n_{output} are the number of input and output tokens in design activity P_i. n_{master} and n_{slave} are the number of master variables and slave variables respectively. From the equation, it can be found that as the value of n_{input} increases, the value of S_i also increases; as the value of n_{output} increases, the value of S_i decreases, which conforms to the token and design freedom relation. Also as the value of n_{master} increases the design freedom increases, which satisfies with the relation of design variable and design freedom.

In MDD graph the initial design activity P_{inital} is a special activity which does not have input design solution tokens defined in this activity. For the initial design activity, because engineers are able to search design solution from all available design space, n_{input} is defined as $+\infty$. Consequently $S_{initial}$ is defined as an equation of the numbers of master variables and slave variables.

$$S_{initial} = \frac{(n_{master} + C_f n_{slave})}{(number\ of\ \mathrm{var}\,aibles\ in\ activity\ i)}, n_{slave} > 0, n_{master} > 0, n_{input} > 0, n_{output} > 0$$

The quality of the overall product design is evaluated using the weighted sum of the quality of each design activity. In this dissertation, importance factor M_i is employed to evaluate the quality of design solution in activity P_i, which is a value from 0 to 1. In the equation of overall design quality, the quality of design is estimated by the weighted sum of all S_i. Because the quality of design is not directly influenced by some special design activities including the activity of finding intersection in Nash strategy, the activity of finding satisfying solution in Pareto strategy and the activity of generating RRS in leader/follower strategy, these special design activities are not considered in the calculation of overall design quality.

$$Q = \sum_{i=1}^{n} S_i M_i , \sum_{i=1}^{n} M_i = 1$$

Based on the design example in this chapter and the design process model in Figure 5.14, the evaluation of the quality of overall design is illustrated in below table. The information that is required to perform the evaluation is provided in Figure 5.14. The Table 5.2 are created by collecting the relevant information in the MDD graph. For the design activity P_5 although it receives 30 tokens from P_3, all these 30 tokens are generated based on the different values of v_6 which is not a design variable in P_5. in this case, the n_{input} and n_{output} for activity P_5 is set as 1. The weights for design activities are set based on the engineers' previous experience. The evaluation result is calculated after proper information is obtained.

Info	Weights	n_{input}	n_{output}	n_{master}	n_{salve}	$n_{variables}$
P_1	0.2	$+\infty$	20	1	2	3
P_2	0.2	20	10	1	1	2
P_3	0.1	20	30	1	1	3
P_4	0.3	30	20	2	2	4
P_5	0.1	30	30	2	0	4
P_6	0.1	20	10	1	1	3
Q=0.2*2/3+0.2*2/3*1.5/2+0.1*2/5*1.5/3+0.3*3/5*3/4+0.1*1/2*2/4+0.1*2/3*1.5/3 =0.4467						

Table 5.2: Information for design quality estimation

The above results Q=0.4467 provides an estimation of the quality of overall design. Multiple MDD graph candidates can be evaluated using the above equations. Based on the evaluation results, it is possible for engineers to choose a proper MDD candidate that is expected to generate quality product design.

5.3.2 Evaluation 2: Overall Design Time

The purpose of calculating the shortest elapsed time for a MDD graph is to estimate how long a collaborative design process lasts. In this dissertation, the time for accomplishing a design process is defined as overall design time. Before overall design time can be estimated, a firing sequence that can fire all design activities in a MDD graph from a given initial marking M_0 needs to be identified. This ensures that the selected firing sequence covers all design decisions. To simplify the calculation of overall design time, there are two alternatives are used in this dissertation: (i) sum up the minimized elapsed time of each a set of simultaneous design decisions or (ii) sum up the maximum elapsed time of each set of simultaneous decisions.

$$TE_{min} = \sum_{n} \min(E(D_n))$$

$$TE_{max} = \sum_{n} \max(E(D_n))$$

$$E(D) = N(D)\,tp(D)$$

In above definitions, D_n is a set of decisions that are fired simultaneously at a time following the firing sequence σ. $E(D_n)$ is the estimated time for each design decision in D_n. The estimated time for each design decision D is associated with the elapsed decision making time tp(D) and the number of tokens that need to be generated N(D). In this dissertation, E(D) is measured as number of tokens generated in design decision D, which is N(D), multiplied by the elapsed decision making time tp, which is tp(D). The elapsed design time tp can be a specific value or a normal distribution. If specific value is selected, the estimated overall minimum and maximum design time can be calculated using the above equations. If normal distribution is selected, the mean value of each design decision substitutes tp(D) in above equation and overall variance of minimal or maximal elapsed time is the sum of the variance of design time for each design decision that is selected as minimum or maximum in the simultaneous decision set D_n. n is the total number of simultaneous decision sets in the firing sequence σ. TE_{min} is the minimum elapsed time that needs to be spent in completing the design process. TE_{max} is the maximum elapsed time that is needed to complete the process. Besides TE_{min} and TE_{max}, the actual design time TE is obtained by running Petri-net simulation. After a specific value or a normal distribution for the design time of each design decision is decided. Petri-net simulation can be executed in Petri-net tools. Many Petri-net software tool provide this simulation function. Based on the analysis of TE_{min} and TE_{max}, if all potential simultaneous decisions are made sequentially, the final design TE is approaching the value of $\sum E(D)$. Hence a measurement of concurrency ratio is defined as:

$$C_r = 1 - \frac{TE}{\sum E(D)} \approx 1 - \frac{TE_{\min}}{\sum E(D)} \approx 1 - \frac{TE_{\max}}{\sum E(D)}$$

In above equation, TE_{min} and TE_{max} are treated as approximate values of actual design time TE. TE_{min} is the design time in the best case and TE_{max} is the design time in the worst case. Using the same design example in Figure 5.14, if the order of marking is {P_1, P_2, P_2’, P_3, P_4, P_4’, P_5, P_6, P_6’}, the firing sequence is available from the initial marking M_0={1,0,1,0,0,1,0,0,1}. This firing sequence can be {(D_1, D_4’, D_2’, D_6’), D_3, D_5, D_4, (D_2, D_6)}. In this firing sequence, there are 5 simultaneous decision sets. According to the definition, if the Pareto, Nash design strategies can be completed without iteration, TE_{min} is calculated as TE_{min}=10+20*30+30*30+30*20+10*10=2210. TE_{max} is calculated as TE_{max}=20*20+20*30+30*30+30*20+20*10=2700. In this simple example, the design time for each design decision is provided as a specific value and TE is calculated as TE=20*20+20*30+30*30+30*20+20*10=2700. If the design time for each design decision is a normal distribution with mean value and standard deviation, TE value can be estimated using Petri-net simulation. Based on the values of TE, TE_{min}, TE_{max} and the sum of design time for each design activity which is 2860 in the example, C_r equals to 0.0559 and is estimated in best and worst cases as 0.2273 and 0.0559, which means the design process is not organized to achieve high concurrency. In mechanical design, if C_r is 1, the design process is organized in a highly concurrent manner. Otherwise there might be a possibility to improve the design process and increase efficiency.

In this dissertation, the elapsed design time for each design decision E(D) is specified based on engineers' experience. There is not an approach to estimate the design times for design activities. Previous experience is important to tell engineers the complexity of a design activity and helps give them a common sense of how long it takes to perform this activity.

The elapsed design time for generating Response Surface or RRS is dependent on the number of design variables in a design activity template and the operating time to perform a design activity template. Based on the information, elapsed design time for generating Response Surface or RRS can be estimated as $3n_vE(D)$ in a full factorial design experiment for quadratic Response Surface model. In the equation, n_v is the number of design variables in the design activity. E(D) is the time needed to solve the design activity template.

5.3.3 Evaluation 3: Design Resource

A Petri-net is said to be k-bounded or simply bounded if the number of tokens in each place does not exceed a finite number k for any marking reachable from M_0. since MDD graphs are used to describe mechanical design process, these graphs are always simply bounded and the number of tokens for each design activity cannot exceed an applicable limitation.

In MDD graph, a token represents a design solution. To generate design tokens, the design resources such as design engineers are required to be involved in to find out proper values of all design variables. For example, in product development, the computation capacity and software license are two types of design resources. And there are some other resources which are needed. Hence the sum of the tokens in a MDD graph indicates the total design resources that may be used in a design process. In general, in a typical design process, the sum of the tokens increases at the beginning of the design works and then decrease as the design goes to a relatively fixed product design solution. The definition of the sum of tokens is provided below.

$$ST = \sum T(D)$$

D is all design decisions. T is the function that calculates the number of tokens generated in each design decision D. ST is the sum of tokens generated by all design decisions from initial marking M_0 to the end when all decisions are fired. An efficient design process is expected to have fewer tokens, which relieves the burdens of designers. Consider the example in this chapter, the sum of token is 123 which is the sum of tokens in all activity P_1 to P_6 and P_2', P_4', P_6' except for some special activities such as intersection and satisfaction activities in Nash and Pareto strategies. If number of tokens is reduced, design resources for product development are saved. However, reducing tokens causes the loss of design freedom in later design activities. If the design freedom is over reduced at the beginning of the design stage, the quality of the designed product is going to be affected. The sum of tokens needs to be adjusted to a proper value. In this case, engineers' experience is important to help set the number of tokens in each design activity.

5.3.4 Evaluation 4: Reachability

Reachability is a fundamental basis for studying dynamic properties of any concurrent system. For Petri-net graph, reachability means that it is possible to transform a Petri-net from initial marking to a specific making. For the mechanical design process, reachability is used to determine whether all design activities can be solved through a design process. The firing of an enabled transition will change the token distribution (marking) in the MDD graph. A sequence of firings will result in a sequence of markings. A marking M, is said to be reachable from a marking M_0 if there exists a sequence of firings that transforms M_0 to M. The set of all possible markings reachable from M_0 in a graph (MDD, M_0) is denoted by R(MDD, M_0) or simply R(M_0).

In mechanical design, engineers are more concerned about a special type of reachability, called the submarking reachability problem which finds if $M' \in R(M_0)$, where M' is any marking whose restriction to a given subset of places agrees with that of a given marking M. The results of submarking reachability analysis in MDD work as a preliminary check to find if there exists a sequence of firing, such that by following this firing sequence all design activities are reachable. For a design process model, at least one qualified firing sequence should be present. Otherwise, it is likely that some activities are missed and never solved. The reachability ratio based on submarking reachability analysis is defined as:

$$R_r = \frac{\text{Max}(P)}{m}$$

in which Max(P) is the maximum number of activities that can be solved following a specific firing sequence starting from the state M_0. m is the total number of activities. Usually this value is 1 and all firing sequences that have a reachability ratio equal to 1 are sequence candidates and ready to be used for scheduling a design process.

For a complex design process model, it is likely to make mistakes and generate a MDD graph which does not satisfy with the reachability evaluation. Consider the same MDD graph of the design example shown in Figure 5.14, if additional arc is added to connect the decision of D_4 to activity P_3, and the input arc weight of D_3 is modified to 3. The process is a deadlock. With initial mark M_0={1,0,1,0,0,1,0,0,1} the process is stuck when P_3 waits for the incoming tokens from the D_4.

5.4 Process Model Selection Based On Evaluation

The evaluation of the MDD graph in Figure 5.14 provides an illustration for performing the evaluation on all MDD graph Candidates. Based on evaluation results, it is possible to select a design process model that best fits the needs of product design. From discussion altogether there are ninety nine MDD graphs that can be generated in the design example of this chapter. To select a design process model from these ninety nine graphs, the four types of evaluation introduced in Section 5.3 need to performed for each graph.

In mechanical design, the engineers have different priorities on the four types of evaluations. Some product developments need to be accomplished in a short time thus they have high priority on the overall design time. Some product developments are asked to save design resources thus consumption of design resource has high priority and fewer engineers are required to implement the design process model. And some product developments need to achieve quality design. In this case, design time and design resource are not as important as the quality of design and it is allowed to take more time to complete product development. With different priorities, the selected design process model is different. In this dissertation, to illustrate the MDD graph selection, overall design time is assumed to be the highest priority.

With the priority on overall design time, the five types of activity grouping in Figure 5.9, 5.10, 5.11, 5.12 are considered to be selected. In Figure 5.16, all different activity groupings are presented. Comparing the groupings in these figures, it can be found if the design time for each design decision does not vary too much, the grouping (2), (3) and (5) in Figure 5.16 can be used to generate design process model with relatively short overall design time. The reason of choosing (2), (3) and (5) is that these groupings are organized in a relative high concurrent manner.

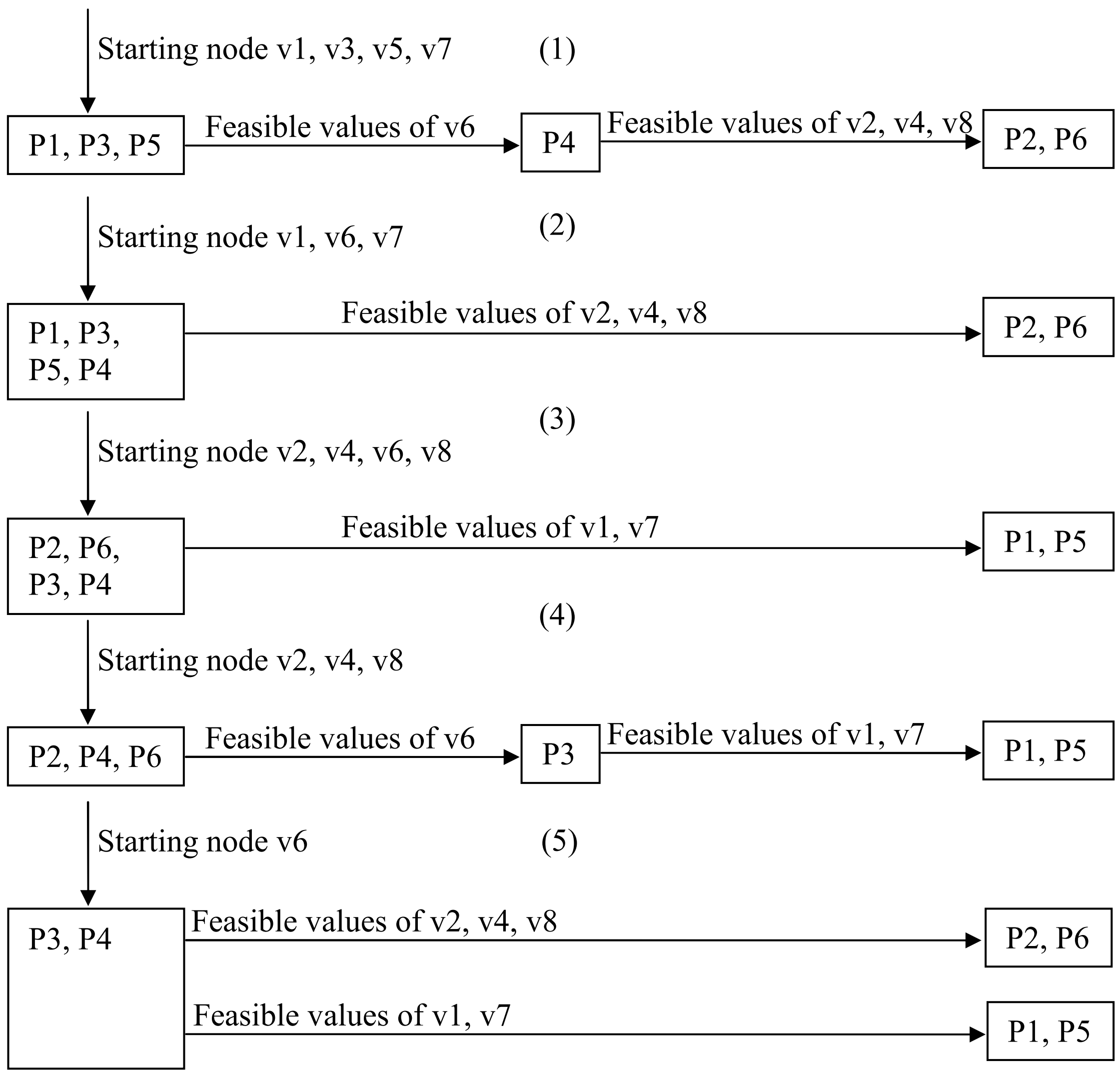


Figure 5.16: Different activity groupings

Based on the same method introduced in section 5.2.2, it is possible for generating different MDD graphs from (2), (3) and (5) groupings. The generated MDD graphs based on above groupings (2), (3) and (5) are illustrated in Figure 5.17. Nash strategy is used to construct design in each MDD graph. In this example, it is expected that Nash strategy is applied to make increase design speed. The evaluation results for three MDD graphs are listed in Table 5.3.

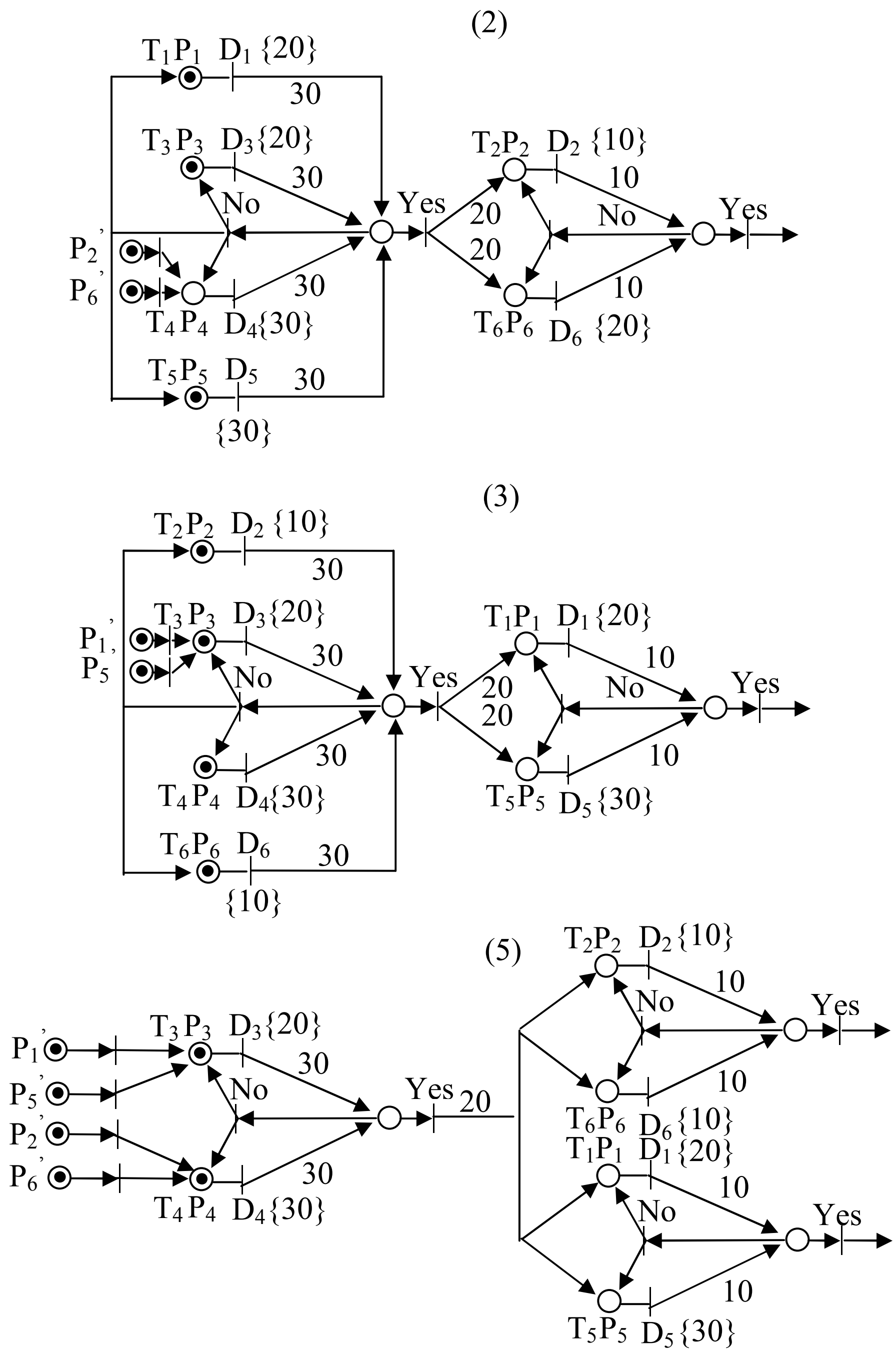


Figure 5.17: Three MDD graph candidates

(2)

Tokens:
T_1-(Mv: v_1, v_3; Sv: v_5 ; RS: $RS_1(v_1,v_3,v_5)$)
T_3-(Mv: v_7; Sv: v_1, v_6; RS: $RS_3(v_1,v_6,v_7)$)
T_5-(Mv: v_5; Sv: v_1, v_3, v_7; RS: $RS_5(v_1,v_3,v_5,v_7)$)
T_4-(Mv: v_6; Sv: v_2, v_4,v_8; RS: $RS_4(v_2,v_4,v_6,v_8)$)
T_2-(Mv: v_2; Sv: v_4; RS: $RS_2(v_2,v_4)$)
T_6-(Mv: v_4, v_8; Sv: v_2; RS: $RS_6(v_2,v_4,v_8)$)

T_4'-(RRS: $v_6=RRS_4(v_2,v_4,v_8)$)
T_2'-(RRS: $v_4=RRS_2(v_2)$)
T_6'-(RRS: $v_8=RRS_6(v_2,v_4)$)

(3)

Tokens:
T_1-(Mv: v_1,v_3 ;Sv: v_5 ; RS: $RS_1(v_1,v_3,v_5)$)
T_3-(Mv: v_6; Sv: v_1, v_7; RS: $RS_3(v_1,v_6,v_7)$)
T_5-(Mv: v_5, v_7 ;Sv: v_1, v_3 ;RS: $RS_5(v_1,v_3,v_5,v_7)$)
T_4-(Mv: v_2, v_8; Sv: v_4,v_6; RS: $RS_4(v_2,v_4,v_6,v_8)$)
T_2-(Mv: v_4; Sv: v_2; RS: $RS_2(v_2,v_4)$)
T_6-(Mv: Sv: v_2, v_4, v_8; RS: $RS_6(v_2,v_4,v_8)$)

T_4'-(RRS: $v_6=RRS_4(v_2,v_4,v_8)$)
T_2'-(RRS: $v_4=RRS_2(v_2)$)
T_6'-(RRS: $v_8=RRS_6(v_2,v_4)$)

(5)

Tokens:
T_1-(Mv: v_1, v_3; Sv: v_5 ;RS: $RS_1(v_1,v_3,v_5)$)
T_3-(Mv: Sv: v_1, v_6, v_7 ;RS: $RS_3(v_1,v_6,v_7)$)
T_5-(Mv: v_5, v_7; Sv: v_1, v_3 ;RS: $RS_5(v_1,v_3,v_5,v_7)$)
T_4-(Mv: v_6; Sv: v_2, v_4, v_8; RS: $RS_4(v_2,v_4,v_6,v_8)$)
T_2-(Mv: v_4; Sv: v_2; RS: $RS_2(v_2,v_4)$)
T_6-(Mv: v_2, v_8; Sv: v_4; RS: $RS_6(v_2,v_4,v_8)$)

T_4'-(RRS: $v_6=RRS_4(v_2,v_4,v_8)$)
T_2'-(RRS: $v_4=RRS_2(v_2)$)
T_6'-(RRS: $v_8=RRS_6(v_2,v_4)$)

Figure 5.18: Detailed information of MDD graphs

After three MDD graphs are generated, detailed information about tokens in each design activity is created by engineers (see Figure 5.18). The evaluations are performed based on these graphs and detailed information. The evaluation methods given in section 5.3 are applied to calculate the results. In table 5.3, for each MDD graph, various evaluation results are listed. From the table it can be found usually one MDD graph does not have better results in all evaluations. The MDD graph based on grouping (2) has shortest design time; the MDD graph based on grouping (3) has the best design quality and the MDD graph based on grouping (5) saves design resources.

Evaluation	Quality of design	Overall design time	Concurrency ratio	Design resources
(2)	0.6389	1120	0.6626	142
(3)	0.7333	1200	0.5472	142
(5)	0.7125	1210	0.4670	104

Table 5.3: Evaluation results of MDD candidates

Selection of MDD graph can be made based on engineers' priorities. If the shortest design time is the most important factor, engineers choose the MDD graphs based on grouping (2) as a suitable graph to implement product development process. Other priorities lead to different selections. A consideration is to calculate a weighted sum to find a suitable MDD graph just like in the House of Quality approach how the proper concept alternative is selected. By setting weights for each evaluation, it is possible to calculate a weighted sum which helps engineers' judgment for MDD selection. In Table 5.4, if the weights for design quality and concurrency are set as 0.4 and 0.6 respectively, weighted sums of three MDD graphs are calculated and from the calculation the MDD graph based on grouping (2) is selected.

Evaluation	Quality of design	Concurrency	Weighted sum
(2)	0.6389	0.6626	0.6531
(3)	0.7333	0.5472	0.6216
(5)	0.7125	0.4670	0.5652
Weights	0.4	0.6	

Table 5.4: Calculation of weighted sum

5.5 Summary

Chapter 5 presents the definition of a Petri-net model that can be used to model collaborative design process. The developed models are applied to describe design processes. The approaches of generating multiple models based on product development information are developed and analyses are provided for selecting the suitable design process that best satisfies with engineering requirements.

CHAPTER 6
TOOLS AND SYSTEM TO SUPPORT DISTRIBUTED MECHANICAL DESIGN

The distributed mechanical design framework presented in Section 3.2 described the major elements including Socio-technical activities, synchronization, communication, coordination, information sharing, process organization and resource utilization, and their relationships in a distributed mechanical design. This chapter presents the requirements of a design support system that can help engineers implement distributed mechanical design according to the framework and a testbed that has been built to satisfy some of these requirements. The chapter is organized to first provide a classification of system requirements, and then present the system architecture and components, followed by the development techniques.

6.1 System Requirements of Building Distributed Mechanical Design System

In order to develop a support system that is ideal and can help realize distributed mechanical design following the framework (Section 3.2), the requirements for the system needs to be first clarified. The overall requirement of the distributed design system is to provide essential management and operation functions through specially designed software or tools to allow designers to perform tasks in a collaborative design process in a distributed environment. This requirement for such a system can be divided into the following categories:

· Requirements to support Socio-technical design activities; (Section 6.1.1)

· Requirements to support dynamic organization and schedule management; (Section 6.1.2)

· Requirements for coordination, synchronization, and communication of Socio-technical design activities; (Section 6.1.3)

· Requirements to support product model management; (Section 6.1.4)

6.1.1 Requirements to Support Socio-technical Design Activities

Engineers conduct different design activities in different design stages. Therefore, various types of Socio-technical design activities need to be supported. Figure 6.1 shows a general pattern of types of Socio-technical activities and their corresponding design stages.

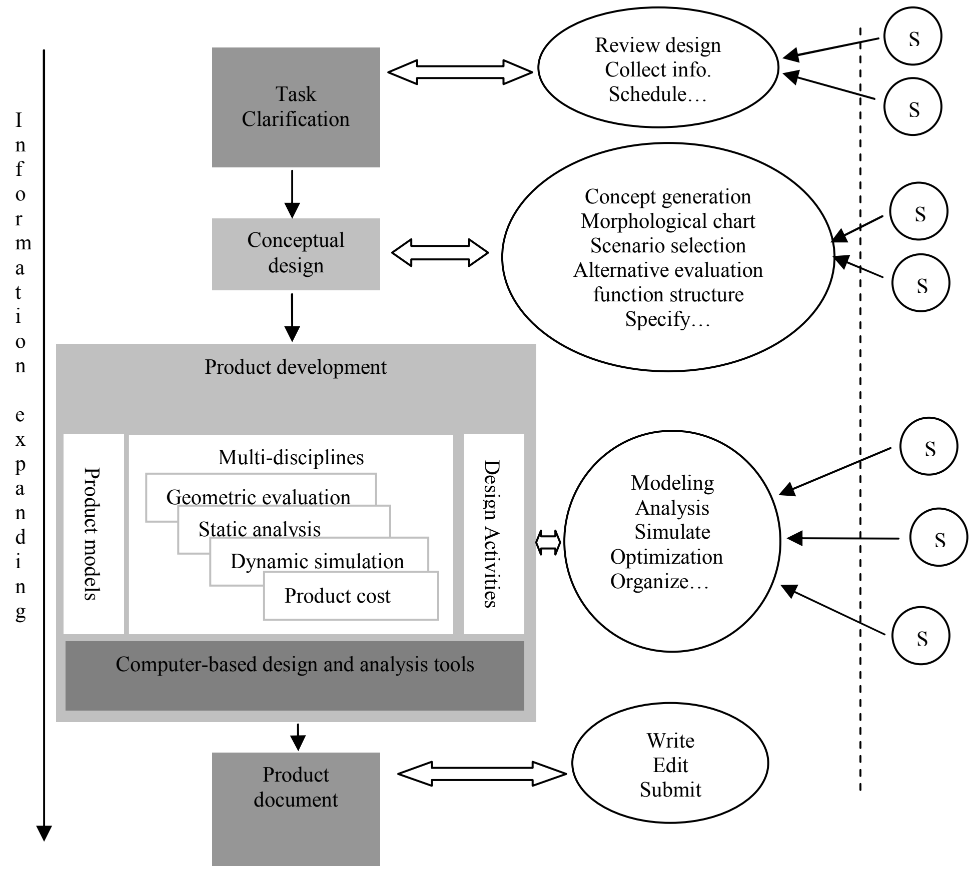


Figure 6.1: Types of Socio-technical design activities

In the figure, the first stage, clarification of task, is an information preparation process. The design activities, during clarification of task stage, mainly consist of technical activities including reviewing previous design histories, collecting relevant information, etc and some social activities including scheduling product design tasks, organizing design teams, making design specifications, etc. During clarification of task design stage, most design activities can be performed efficiently even by individuals. Synchronization of engineers' technical activities on multiple sites is not required. Communication, especially multi-user group discussions, is the main form of collaboration between engineers. Support for effective communication during social activities is the requirement of the design system in task clarification design stage. From conceptual design stage more engineers are involved in the design process. Conceptual design is a stage that can greatly influence further design works and in some degree decide the final product design quality. Although it is important, so far not many effective collaborative design tools have been developed. Conceptual design is still largely dependent on designers' knowledge and individual activities. Collaboratively conducting design activities in this design stage is considered an important approach to improve design. In conceptual design stage, various technical activities are needed, including concept generation, function structure generation, alternative evaluation, selection of concepts, etc. In addition, social activities (i.e. specifying evaluation criteria for product performance, exchanging knowledge and experience) need to be performed. For the design system, in conceptual design stage supports for real-time synchronization during each type of the technical activity and communication (i.e. discussions, instructions, notifications) for social activity are required. The next phase in the design process, product development, requires intensive computation. Multiple computer-based analysis and simulation tools are utilized to complete this step. In this stage, CE is widely accepted as an effective principle to speed up the development process and reduce the development cost. In product development design stage, most of technical activities, including modeling, simulation, analysis, optimization, and making design decision, are supported by computer based software. Most of these software tools are used by engineers to complete their design tasks individually, and does not require extensive collaboration. Social activities in this stage are not frequently performed. Engineers follow the guidance or advices, which are specified in the previous conceptual design stage. Synchronization interaction is the main form of collaboration. Models and data created in this stage need to be synchronized in real-time or non real-time to ensure data consistency. The requirement to the design system in product development stage is that the system supports real-time and synchronized design activities, such as collaborative modeling, analysis and simulation. The last stage, product document stage does not show significant need for collaborative design activities and is not required to support multi-user collaboration.

The requirements of the distributed mechanical design system include facilitating engineers to collaboratively perform technical and social activities in each design stage and to support essential synchronization, coordination and communication interactions. Table 6.1 lists design phases and corresponding collaborative design activities that need to be supported.

Design phases	collaborative design activities
Customer requirements and product specification	Technical: usually only individual behaviors Social: scheduling, plan, organization, specification
Conceptual design	Social: specification, selection Technical: • Generating concepts • Generating morphological chart • Selecting scenarios • Evaluating alternatives • Generating function structures
Product development	Social: not frequently performed cooperatively Technical: • modeling • analysis • simulation • optimization • …
Product document	Social: not frequently performed cooperatively Technical: not frequently performed cooperatively

Table 6.1: Summary of requirements for Socio-technical design activities

6.1.2 Requirements of Dynamic Organization and Schedule Management

In the framework of Section 3.2, organization and scheduling are important elements of the design process. One of the requirements for the framework includes providing proper privileges to design team members based on certain types of social roles, which can help engineers to dynamically set up a design organization and task schedule for management of the design process. The changes of organization and schedule can later influence engineers' social and technical roles and thus result in different Socio-technical activities. In order to successfully support an efficient design process, the design system should provide basic functions, such as user management, allowing only permissible users to perform administrative works, and task management, assigning design works to relevant users. Based on the user management, the collaborative design system should facilitate the manipulation of the information about engineers' professional background, offered privileges, position in organization, etc.. Based on the task management, the design system needs to facilitate the manipulation of the information about work assignment, overall schedule and responsibility of engineers.

In Chapter 5, Model of Distributed Design was introduced. The MDD graphs provide information about how design processes are organized. Based on MDD graphs, the workflow of activities, responsible engineers, and schedule of design tasks are planned and configured before the actual design activities start. To accomplish successful collaborative mechanical design, it is necessary to support engineers with software tools to generate alternative models for design process, manage design resources, and product development schedule. The design system should be equipped with various process management tools.

6.1.3 Requirements of Synchronization, Coordination and Communication

Synchronization, coordination, and communication are important elements of the distributed mechanical design framework. The synchronization, coordination and communication make it possible for multiple engineers to interact with each other based on their unique design perspectives. Therefore engineers in a distributed design environment can work collaboratively.

In this dissertation, coordination is the interaction through which engineers share their design activity information, make decisions in aware of other engineers' possible responses. In order to support efficient coordination, some requirements for the design system must be satisfied. The support of transferring design solution in a uniform template is the first requirement. Since design solutions are used for engineers to express their design decisions, the design system should record and share this information with engineers who are responsible for dependent design activities so that engineers can easily capture design intent of various disciplines and make suitable design decisions. The transfer of design solutions can be handled by file servers and computer network. Engineers login the servers to access the relevant design decision information in product development.

Synchronization is the interaction through which the working objects or other engineering data in the workspaces of multiple engineers are updated and kept in consistent. In order to achieve efficient synchronization, the design system is required to manage the access of data to avoid inconsistency and real-time update of the information in multiple sites. Client server structure works well to manage data in multiple sites by keeping the information on clients consistent with the data on the server.

Communication is the interaction that corresponds to multiple engineers chatting, talking and writing to each other to share their design views in the product development process. Commercial messenger software such as MSN can support a group of designers to communicate with each other. As for mechanical design, a very important factor to achieve effective communication is to involve product model and other engineering data into the communication process so that engineers can understand each other better. Communication based on mechanical design models and data is a requirement for an ideal collaborative design system.

6.1.4 Requirements of Supporting Product Model Management

Product data and model should be managed by the design system so that with the proper role based privileges, an engineer can store, access, share, and transfer a model or file with other engineers. In order to support the above file operations, some requirements for the design system are: (i) The product data and model should be managed in a structure through which a designer can easily find the product data; (ii) Basic file access controls are required to keep the data secure and restrict illegal operations; (iii) The design system is required to provide some basic functions to manage the file operation privileges based on designers' Socio-technical roles; (iv) Various file status needs to be supported by the design system to prevent the file inconsistency which may occur when two users modify the same file simultaneously.

6.2 Distributed Design System Architecture

The system architecture and components for a a basic collaborative distributed design system are introduced in this section. Functional tools, which are the building blocks for the design system, are presented in Section 6.2.1.

Enabled by online web interface, the collaborative design system architecture supports distributed users collocated, within the design environment, collaboratively anticipating design process through a web-based design system. The proposed architecture consists of four functional servers: (i) GDS/software integration server, (ii) web server, (iii) product model server and (iv) database server.

As illustrated in Figure 6.2, a web server is developed to generate the dynamic HTML interface for engineers. Since different engineers are responsible for different roles, the design system supports personalized management and adjusts the user interface according to their Socio-technical roles. Through user login, a designer is provided with the proper information to perform various design activities, considering interactions with other engineers and the management works such as product model management, organization and schedule management, etc. The information that is needed by the web server is managed by a backbone database server which is also shared by two other functional servers, GDS/integration server and product model server.

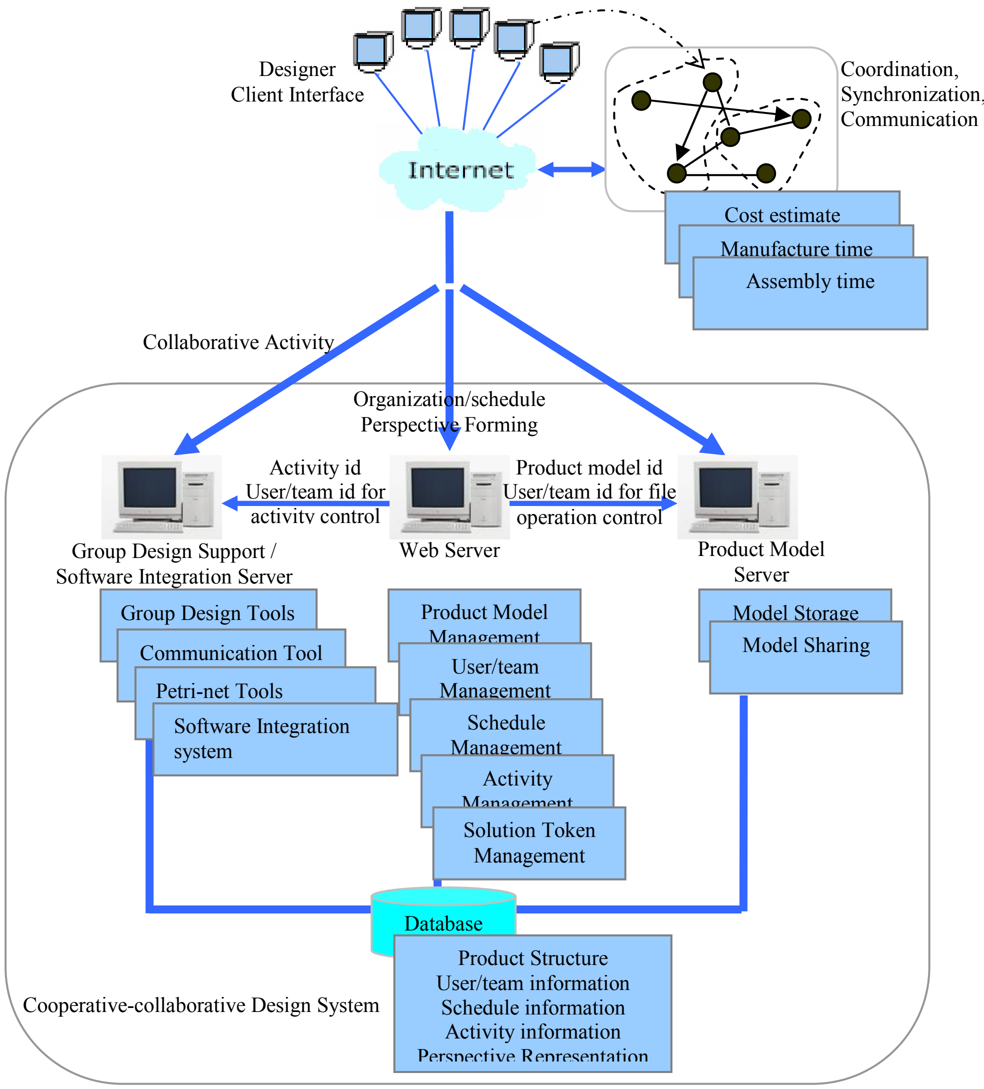


Figure 6.2: Architecture of Collaborative Design System

The GDS/integration server supports the real-time collaborations of engineers. In the developed design system, the GDS/integration server includes Group Design Tools such as the concept generation tool, function structure, alternative evaluation tool, selection tool, , etc. Most of the Group Design Tools are developed to support multi-user real-time design activities from concept design stage to product development stage. Besides Group Design Tools, Petri-net tools are integrated into the system to help engineers create MDD graphs and organize design process. The software integration system and communication tool are also provided by the GDS/integration server. Software integration system is designed to integrate commercial software into the design system. It is important for multi-users to collaboratively perform detailed design work using commercial tools. The communication tools are the general tools available for engineers to exchange their ideas and opinions. The tools will launch various types of communication between group members such as email, notification, text chat, audio/video conference, model discussion etc. and let engineers exchange the information about the product design. Among the above communication types, model discussion is an online real-time cooperative method for designers to check geometric models and attach some comments for better understanding each other's considerations.

The product model server is used to store the product information, CAD models, and other documents generated in the design process. After an engineer creates some product data, it is natural for him/her to upload the files into a file server and let the server automatically manage the files and store them into specific database tables. The product model server also supports data sharing for multiple users. The control of the file status avoids errors that are usually occurred in a multi-user read/write mechanism.

The database server provides the data access for other servers in the design system. In order to successfully run the functions of the distributed design system, various servers need data access supports. For example, the web server needs the user and team id in the database to generate dynamic user interface for displaying the user relevant perspective information in the system. The product model server needs the configuration information and product model id to decide where the submitted files should be stored. The GDS/integration server needs the user and team information and activity id in the database to manage the group design activities.

6.2.1 Essential Components of Cooperative-collaborative Design System

From the introduction to various servers, five major components of design systems can be identified:

Group Design Tools:

This component of the design system includes a set of real-time and multi-user cooperative design tools, which are developed based on the requirements of cooperative design in conceptual design stage. The tools developed are design-oriented and consider the special needs for designers in various Socio-technical design behaviors.

Software integration system:

This component includes some software wrappers that integrate the commercial design tools into the system. Through software wrappers, engineers are provided with the ability to use the distributed heterogeneous design resources so that engineers have more tools to select.

Process management system

The process management system is developed to provide an entrance for engineers to use basic design system functions and tools. It works as the starting point for users to find what he/she needs and directs them to use the system functions, such as task and user management or Group Design Tools with proper control of his/her privileges.

Petri-net Modeling Tools

Design process organization is assisted by Petri-net modeling software which includes basic model creation and advanced simulation based on created models. A set of Petri-net tools are available that can be used to perform this task. Among them, WinTPTPN is a tool that is capable of running various analyses for Petri-net.

Product model management system

The product model management system is developed to store product model and other engineering data and share useful design information among engineers. Through a web based interface of product model management system, engineers can upload and download design information into the product model server.

6.3 Implementation Techniques

In order to implement the above design system, various development techniques are needed. This section presents a brief introduction of most techniques that are used in the development of the system. The introduction is categorized into three groups, namely web based programming, database application and middleware infrastructure.

6.3.1 Web Based Programming – PHP, Javascript, Java, Cortona SDK

Most management functions are implemented by web based techniques. Considering the application characteristics and development convenience, different development tools are chosen for implementing different functions. In this research, PHP is used for generating dynamic role based user interface. Javascript is widely used for various purposes from handling user events, improving user interface, to operating on plug-in components, connecting with Java applications and so on. Java is mainly used for client-server communication and user interface. Cortona SDK is used for 3D VRML programming.

PHP is a server-side scripting language and interpreter that is available on a wide range of platforms, including some versions of Apache, and Microsoft's Internet Information Server (IIS). The original program was called Personal Home Page Tools, which is where the initials PHP come from. The PHP language is used to develop product data management system and process management that are introduced in Section 6.4.

Java is an object-oriented language similar to C++, but simplified to eliminate language features that cause common programming errors. Java source code files are compiled into a format called bytecode, which can then be executed by a Java interpreter. Compiled Java code can run on most computers because Java interpreters and runtime environments, known as Java Virtual Machines (VMs), exist for most operating systems, including UNIX, the Macintosh OS, and Windows. Jave is used to develop 3D model discussion tool in Section 6.4.

JavaScript is an easier to understand, less complex version of its distant cousin, Java. Developed by Netscape, it carries with it a smaller command set and a much simpler structure, though it remains an OOP (Object Oriented Programming Language). OOP can make a language easier to tackle, by breaking a program up into 'parts' to make up the whole. Something important to note is that JavaScript is unable to stand on its own like Java. It is a text-based language that must be placed within HTML, to be read by the browser and interpreted so the instructions can be performed. Javascript is used to handle user input events in the online applications in Section 6.4.

Cortona SDK provides an application programming interface (API) that enables authors and developers to integrate ParallelGraphics 3D technology into other applications developed by Visual C++, Visual Basic, Delphi, or third party applications supporting ActiveX technology (like MS Access, MS PowerPoint) as well as HTML and Java applications. The documents describe the objects, properties, and methods exposed by the

Cortona ActiveX control. Cortona is used to manipulate VRML models in 3D model discussion tool in Section 6.4.

6.3.2 Database Application - MySQL Database Server

In order to store, query and modify the data for the use of our web server. A database system is needed. For web based application, a widely used database system is MySQL which is free for developing non-commercial software.

MySQL is a Relational Database Management System (RDMS). Using a RDMS means it is possible to add, access, and process the data stored in their database. 'SQL' stands for "Structured Query Language" - the most common standardized language used to access databases. MySQL is Open Source software and is freely available at www.mysql.com. As a popular database software, many online applications are developed based on the MySQL database.

6.3.3 Middleware Infrastructure

In this section, a brief introduction of the implementation techniques that are used for software integration in our design system is presented. The emergence of distributed applications has raised the need for portability across numerous software and hardware architectures. A way to address this problem is to use a middleware when designing a new distributed application [117]. Middleware is a uniform infrastructure that can be used by engineers to design distributed applications. By ensuring compatibility of middleware to most of operating systems and programming languages, software built above the same middleware can link and interoperate with each other. One popular commercial middleware is CORBA which is a specification adopted by Object Management Group (OMG) [118].

6.4 Implementation of Distributed Design System

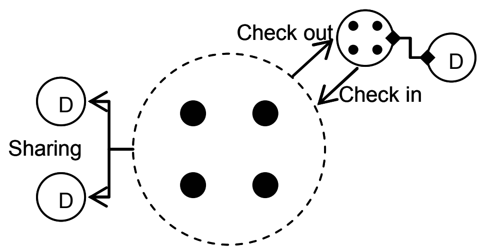
In this dissertation, three major components of the distributed design system, Group Design Tools, process management system, product model management system are developed. The rest of components are adapted from existing systems or software. The Group Design Tools can be provided by developing new software or by wrapping existing software. Using the first approach, the collaborative design features are developed in the product and software can obtain maximum performances. Some software companies, such as PTC and Unigraphics, which are vendors of solid modeling and Product Data Management (PDM) software, already demonstrated some prototypes of such systems. However commercial products with full collaboration supports are still not available in the market. Another approach is to develop Group Design Tools based on the existing software. Compared to the former, this approach requires less effort and time. It also provides compatibility to the legend systems. In this dissertation, some Group Design Tools [119] for conceptual design stage are designed using the first approach, because in conceptual design stage, not too many commercial tools have been released.

Besides the Group Design Tools, the process management system are developed using web based programming and database techniques. Synchronization and communication systems are based on TCP socket and programmed using popular language such as Java or C++. The following sections are organized in this way. In section 6.4.1, before the detailed introductions to systems and tools, models of synchronization in design tools are first given. In section 6.4.2 and 6.4.3, introductions of process management system and other Group Design Tools for the conceptual design stage are presented respectively..

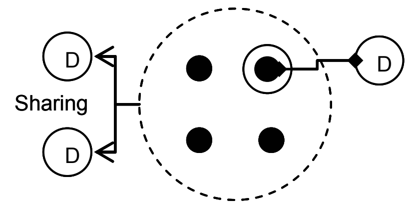
6.4.1 Models of Synchronization and Communication

In order to develop various Group Design Tools, product model management system and process management systems, it is essential to illustrate various scenarios of synchronizations. The requirements of synchronization models for design information, process and product management information are the same, critical data need to be synchronized in real-time and ordinary data need to be synchronized in an allowable period to reduce out of date data. Illustrated in Figure 6.3, Models 1-4 illustrate the synchronization in various software applications.

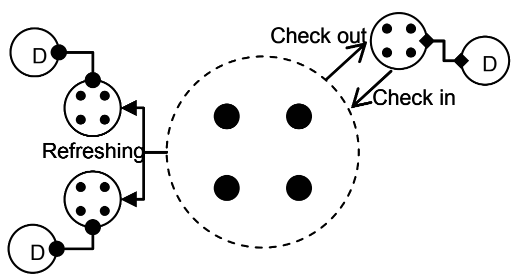
In a communication application, information is shared using channel, multiple users receives and sends the information using the same channel. Model 5 in Figure 6.3 illustrates a model of communication in a messenger application.



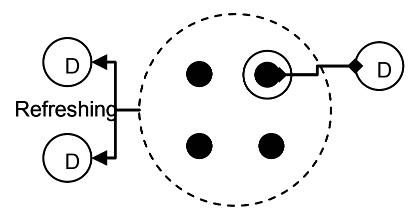
Model 1: Local modification with delayed synchronization check out and check in and global viewing of objects with delayed synchronization sharing.



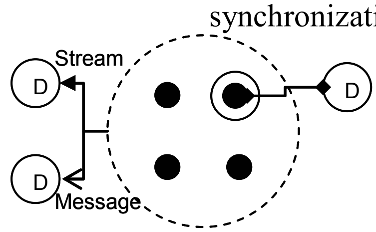
Model 2: Direct modification of partial of global objects with access privileges and global viewing of objects with delayed synchronization sharing.



Model 3: Modification of checked out local objects and viewing checked in changes with refreshed local objects.



Model 4: Direct modification of global objects with access privileges and viewing of global complete objects with real-time synchronization, refreshing.



Model 5: Publication of media information by a channel and receiving of channel information with real-time communication stream and non real-time communication message.

Figure 6.3: Models of synchronization and communication

Real-time and non real-time synchronization are essential and need to be combined to provide different levels of interaction (Figure 6.3). The models of synchronization are implemented in various Group Design Tools or management systems. In Figure 6.3, dashed circles are used to represent the range of working objects that can be reviewed by a designer and solid circles represent the range of working objects that a designer has the privileges to manipulate. The dark circles represent working objects. Four dark circles combined represent all sets of working objects. Solid arrows represent the real-time synchronization of working objects and hollowed arrows representing non real-time synchronization of working objects. Arrows with dark box ends represent the operations to perform design changes by designers. Arrows with dark circle ends represent the operations to view the information of working objects without any changes. Circles with a capital "D" represent the designers in cooperation.

Model 1 represents the situation in which all designers need to check out working objects to a local computer before modification and check in objects for others to share design changes. The global objects are locked when one designer checks out the objects for the operations. As a result the changes of the working objects are not known by other designers until the working objects have been checked in and another round of sharing the global objects have been launched. The types of synchronization such as check out, check-in and sharing in Model 1 are not real-time. Model 1 depicts the interaction which is based on delayed synchronization and widely used in the applications for product data management. Model 2 represents the situation in which designers can directly modify only parts of complete objects with the given access privileges but can share whole information of global objects with others. The information sharing in this model is a delayed synchronization. Designers need to login and check for the latest design changes. Model 2 is often used in applications of process management. In a process management such as adjusting organization, only designer with proper access privilege can adjust the organization data in database and each designer needs to be login to check his/her new position in the organization. Model 3 represents a situation where all designers can view the complete synchronized local objects through real-time synchronization, refreshing, which updates local objects according to global objects in case any changes occur. Model 3 can achieve efficient synchronization because designers only operate on local objects. The shortcoming for this model is the heavy network traffic due to simultaneously updating local objects by refreshing. In this proposal, 3D Model Discussion tool and Concept Generation tool apply the Model 3 synchronization. Model 4 represents a situation where designers directly modify partial working objectives with proper access privileges and view entire objects in real-time with refresh synchronization in case any changes occur. Model 4 is widely discussed in applications such as collaborative CAD modeling tools, in which multi-users attempt to modify geometric models with proper privileges and at the same time view others' design changes.

At the bottom of Figure 6.3, Model 5 is presented to illustrate the model of multimedia communication. In Model 5, dark circles represent the information channels. Solid arrows represent real-time communication type, stream. And hollowed arrows represent non real-time communication type, message. Senders are required to publish their media information by a channel and receivers receive the multimedia information via real-time stream or non real-time message communications.

6.4.2 Process Management System

The design process management system is a task based design portal (see Figure 6.4) that has several user levels for designers: project management, team management, and task management. The system also has the ability to manage and monitor a design process for different groups. The development of design process management system follows the synchronization Model 1 and Model 2 in Section 6.4.1.

The project manager usually initiates the project management for each development project by assigning a project space and then adding users to different design groups. Once the project space and the groups have been formed the design team manages the design process for the project. Functions provided by the system for project administration includes:

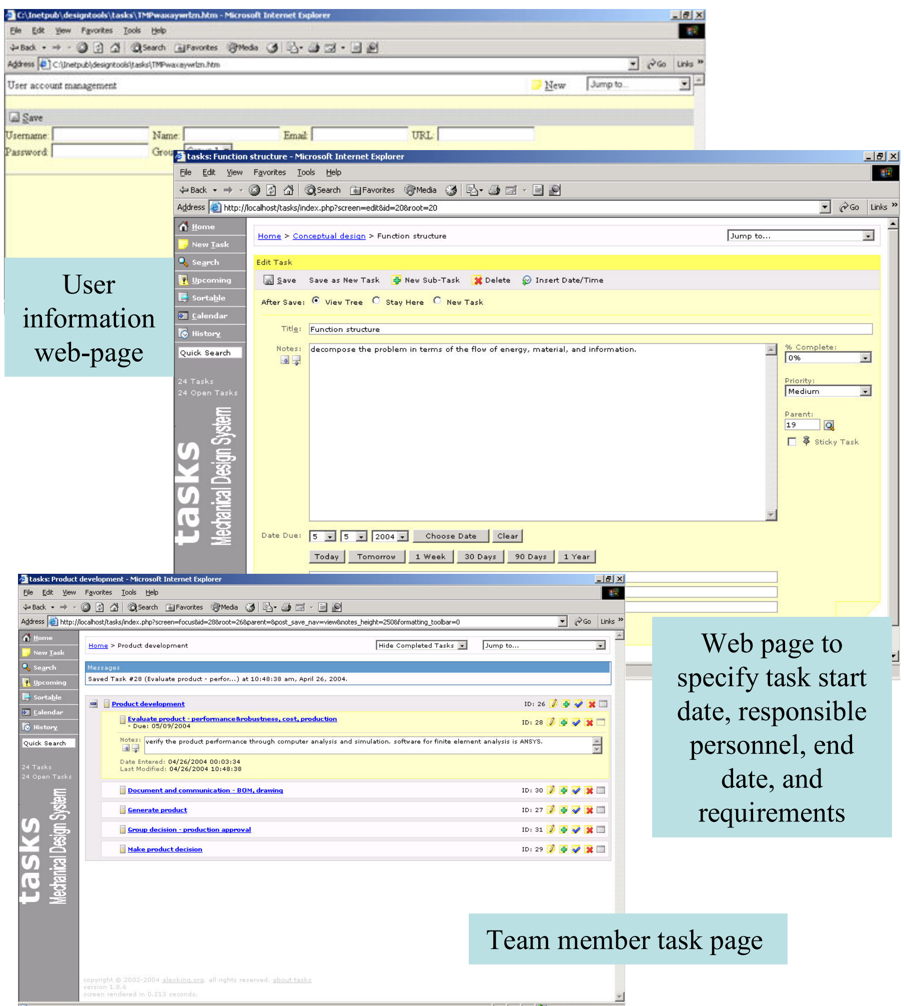


Figure 6.4: Task based design process management system

- *User account management* involves specifying organization of the team and assigning responsibilities for team members. The design team is organized to facilitate planning and coordination for achieving high productivity, which is accompanied with individual responsibilities.
- *Design process template loading and configuring:* A general design process template can be loaded by the manager for each design team. The design process template should be based on general steps followed by the company during product design and development. The team uses the template to design their team design process for the project through modification, addition, deletion and division of tasks into more specific tasks.
- *Task assignment:* Responsibilities for the tasks for the design project are assigned to different members of the team. Estimate date for start of each task and time required to complete the tasks are also specified. All these information can be first decided by the selected MDD graph
- *Task description and requirements:* The designers, as a team, describe the objectives of each task, along with their requirements. These general information can be imported from the MDD graph which contains more detailed information in its activity templates.
- *Sub-system interface:* This function allows designers to specify critical sub-system interfaces and parameters. Once specified, the system notifies appropriate team members if changes to relevant sub-system interface are made.
- *Rescheduling:* This function allows managers to reschedule and redistribute design tasks and assign new date/time to complete different tasks.

Functions supported by the system for designers to complete tasks include:

- *Obtain task information:* This function allows design team members to obtain information related to tasks assigned to them. Tasks that need to be started and tasks that are approaching deadlines are placed in high priority category.
- *Built-in design task contextual help or step instructions:* Easy access to help and information on how to accomplish the task and tools to accomplish the task are provided.
- *Online file submit system:* This function allows uploading of information for the project so that it can be accessed by other members of the team.

6.4.3 Group Design Tools for Conceptual Design

A set of design tools for conceptual design stage have been developed and a description of the tools is introduced below.

6.4.3.1 House of Quality Tool

The House of Quality is an approach that can be used to understand customer requirements and specify engineering targets for the project. Building the House of Quality is a collaborative design activity that requires extensive discussion and considerations. Supported by the House of Quality tool, a group of designers can login the same session and develop the House of Quality cooperatively (Figure 6.5). The communication among the different designers using the House of Quality tool is handled by the real-time multimedia communication tool developed using Macromedia Flash Communication Server. The synchronization Model 3 in section 6.4.1 is applied in this tool. The House of Quality tool creates a shared object on the server, for any change in the data entries. The shared object stores the data and informs the server with the new changes. The server then sends the event of data change to all the clients. The client side receives the event sent by the server and refresh local data according to the code inserted in the event handler. As a result all users have the same view of the House of Quality as information is added/modified. In addition, text chat and audio/video stream are added at the bottom of the House of Quality webpage to facilitate discussion among the team members.

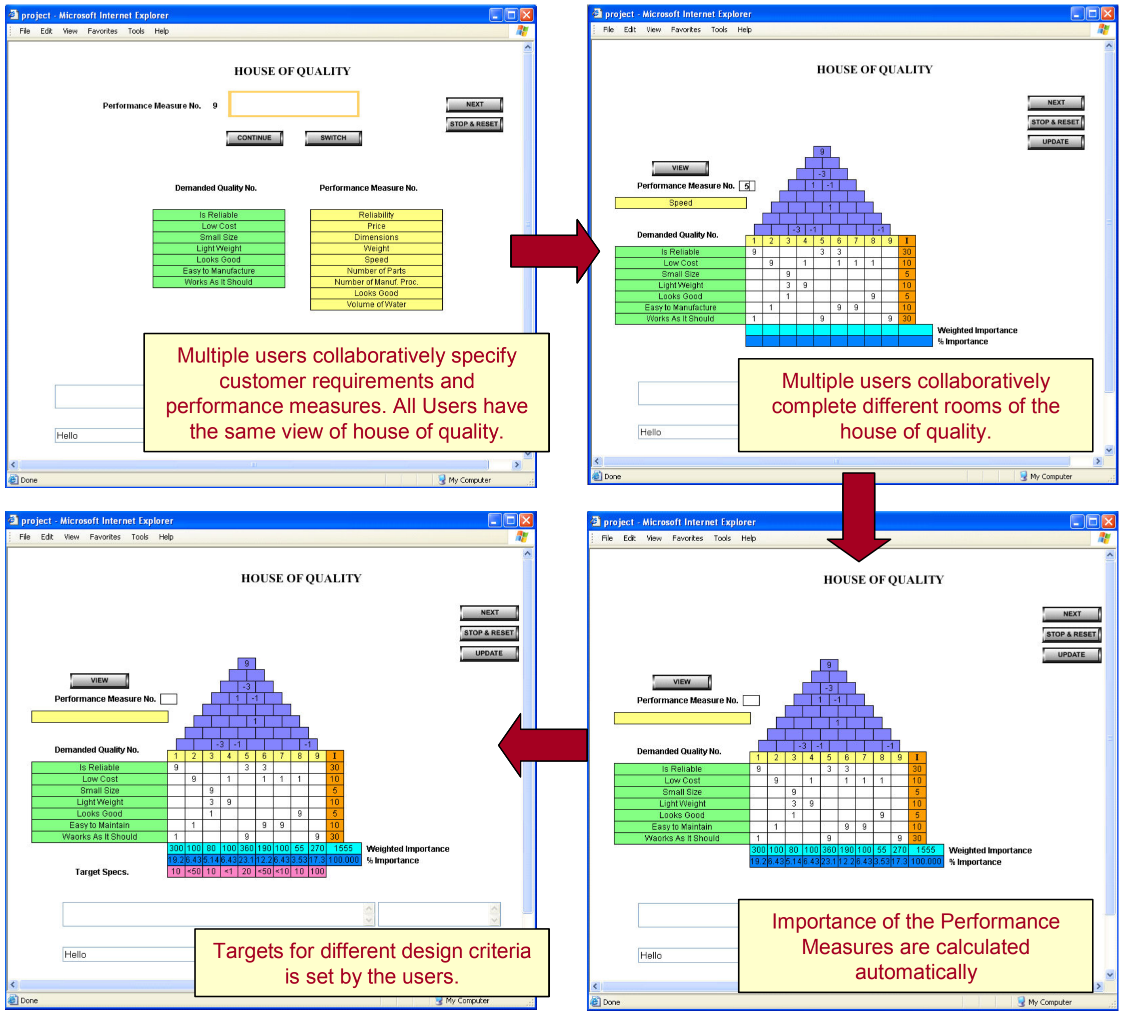


Figure 6.5: Collaborative creation of a House of Quality by group designers

6.4.3.2 3D Model Discussion Tool

Discussion of concepts using 3D geometry is an essential technique for cooperatively generating design concepts among designers (see Figure 6.6). To enhance cooperation among different members of the team, the cooperative 3D model discussion tool allows users to add notes to the 3D geometric model, and exchange text and audio information in real time. By loading and sharing a 3D geometry, users have a virtual environment to view some basic structures and layouts of the product. The preliminary information that a user obtains from a 3D model discussion tool is helpful for a brainstorm process of finding all possible design concepts. In this proposal 3D model discussion tool is developed with some useful characteristics. First, the tool is designed to provide a real-time synchronized users' viewpoint. This characteristic is important because in a collaboration process designers need to make sure that they are talking about the same part of model. The second characteristic is our 3D model discussion tool supports inserting text comment into geometric model. With this function, a user can easily attach a comment to one point of the component so that others can clearly get the right instructions. The viewpoint adjustment and comment insertion are managed by a synchronization server. All shared data are stored in this server. Client application receives the change from the server or requests a change to server data by socket connection. Server application locks the shared data in the process of synchronization to keep the data integrity. This tool applies the synchronization Model 3 in section 6.4.1.

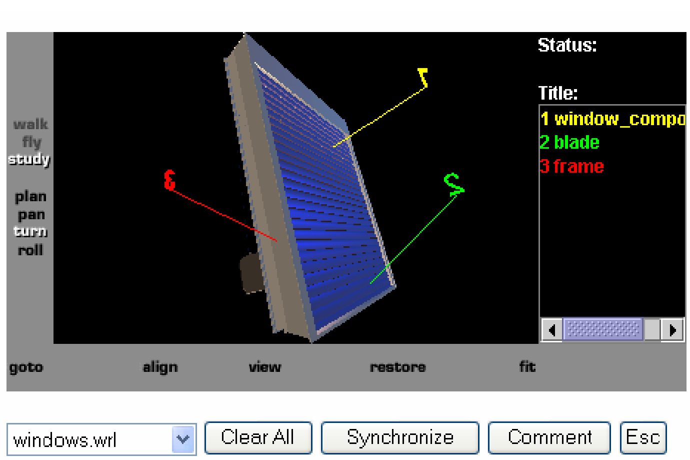


Figure 6.6: Cooperative 3D model discussion tool

6.4.3.3 Concept Selection Tool

The web alternative selection tool follows the basic steps required to perform selection (Figure 6.7). First information related to all concepts needs to be entered through the "Alternative Information Sheet". Information related to each concept can be entered by a team member. Means for text and audio chat are provided by multimedia communication tool. Once the information, including sketches, has been entered, it can be accessed by any member of the team. Next the evaluation criteria or attributes need to be identified, along with the associated relative importance (weight) and evaluation scales. The attribute information, using the web alternative selection tool, is specified cooperatively by a team. Any information entered in the attribute page by a designer is automatically transmitted in real time to all designers participating in the selection process. With the alternatives and attributes for the selection process specified, the next step is to cooperatively provide rating for each concept for each attribute using developed scales. Once all information has been entered in the "Rating Page" calculations are performed and a text report is generated, which can be used in a design report. The communication among the different designers during the process of using web alternative selection tool is handled by multimedia communication tool with support of Macromedia Flash Communication Server.

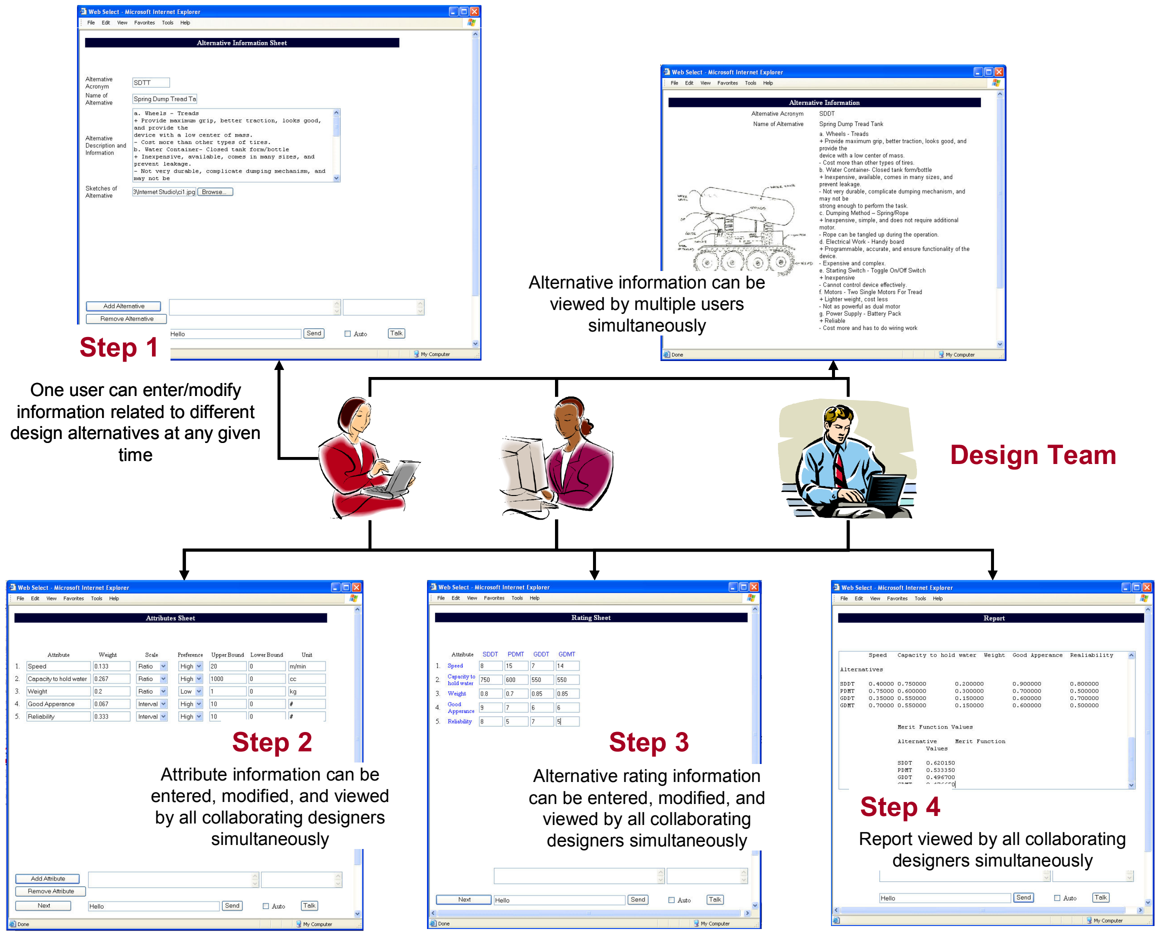


Figure 6.7: Cooperative alternative selection tool

6.4.4 Multimedia Communication Tools

Discussion, instruction, learning and a lot of Socio-technical activities needs the support of multimedia communication. In this design system, the communication among the different designers is handled by multimedia communication tools developed using Flash Communication Server. Based on the cooperative mechanism in FCS, our multimedia communication tools have the abilities to check user account, arrange users in different web conference room based on shared objects and publishing text/audio/video information by media channels. (see Figure 6.8) The multimedia communication tools are important cooperative tools which help designers in distributed environment exchanging the understanding of product design. Since the multimedia communication tools are used in almost all design activities in a design process, the multimedia communication tools usually work accompanying with all other cooperative tools.

Figure 6.8: Multimedia communication tools

6.4.5 Petri-net Tools

The Petri-net tool used in this dissertation is WinTTPN that supports Petri-net model creation, simulation and analysis. The software WinTTPN provides graph user interface for engineers to add Petri-net place, transition and connection. For each transition, the elapsed time can be set as a normal distributed random value. The simulation based on Petri-net can give important information such as the design time required to complete the design process.

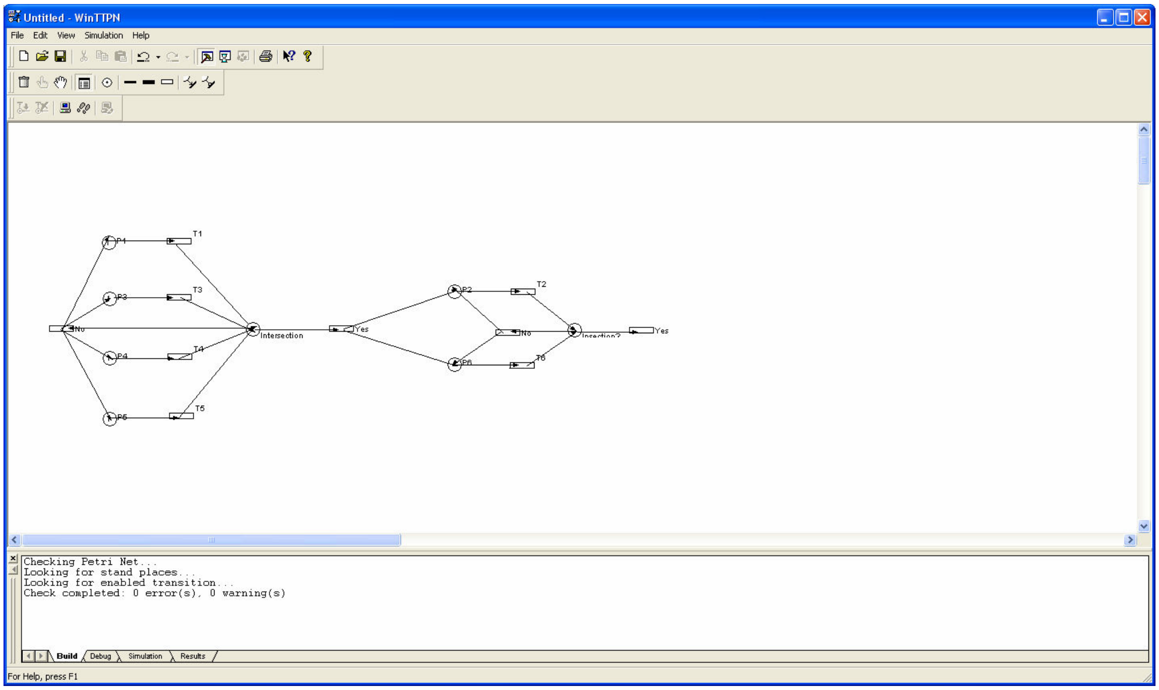


Figure 6.9: User interface of Petri-net software WinTTPN

Figure 6.9 is a screen shot of one created MDD graph using the software WinTTPN. From the figure, it can be found that creating and running analysis using Petri-net tools are convenient for engineers.

6.5 Summary

In this Chapter, an introduction of the system requirements and implemented system and tools are presented. Section 6.4.1 presents the models of synchronization that need to be implemented in various design tools. Sections 6.4.3.1 to 6.4.3.4 introduce the functions of process management system, House of Quality tool, model discussion tool, alternatives selection tool and communication tools. These tools are the building block and are integrated in the distributed design support system.

CHAPTER 7
APPLICATION OF DISTRIBUTED MECHANICAL DESIGN

In Chapter 7, two application examples of distributed mechanical design are presented. These examples illustrate how collaborative design approaches and design support system are utilized in product development. The first example is a light weighted manipulator product design. In this example, the design decision making approach and MDD graphs introduced in Chapter 4 and 5 are utilized to help engineers organize the design process and find design solutions satisfying with the requirements from multiple engineering disciplines. The second example is a reverse engineering project. The focus of the second example shows once the design process is planned how design support system can help engineers go through the design process and complete design works with quality product designs. The design work and analyses shown in the second example has been performed by Prof. Chang and his students. We thank Prof. Chang and his design team for their extensive help and cooperation [124-126].

In the following of this chapter, the two examples are included in section 7.1 and 7.2 respectively.

7.1 Example of Light Weight Manipulator Design

An application example of distributed mechanical design is light weighted manipulator design. In this dissertation, engineers from different disciplines have different considerations about what kind of product they want. In this section, design decisions based on basic considerations of each discipline are discussed to explain how these distributed engineers collaboratively complete mechanical design. The structure of the manipulator is illustrated in Figure 7.1. The manipulator consists of three arms and three joints with 3 degree of freedoms. A Total of 9 parameters need to be determined. These parameters are radius r_1, r_2, r_3 and lengths l_1, l_2, l_3 of three arms, and angular positions θ_1, θ_2, θ_3 of three joints. In this design example, engineers from three disciplines work together. These engineers are static engineer, geometry engineer, and cost engineer. During the design, engineers in different disciplines will have different performance considerations of the manipulator product. Static engineer attempts to minimize the manipulator weight; geometry engineer attempts to achieve a large clearance space under the manipulator arms and cost engineer needs to control the manipulator cost. The overall c-DSP formulations for the three design activities are provided in Figure 7.2.

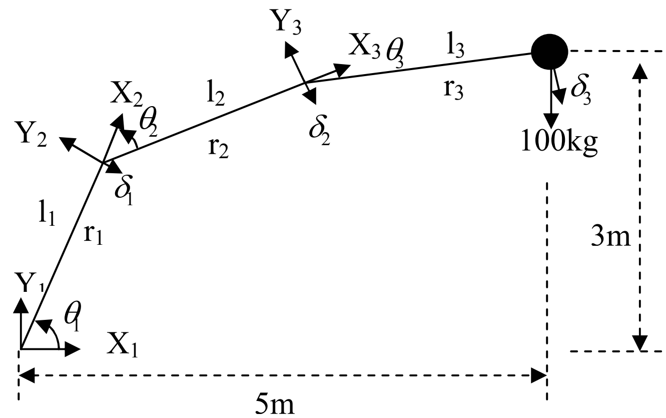


Figure 7.1: Manipulator parameters

In this dissertation, the illustration of how to use the distributed design approaches to perform distributed product development is presented in three steps. The first step is to define all design activity templates. The second step is to organize the design activities into design process and improve design process using MDD graphs. The third step is to implement the product development with the help of various design tools and design support system.

7.1.1 Design Activity Template

In Figure 7.2, the static engineer's consideration is to minimize the weight of the manipulator with the deflection constraint. In this example, the constraint is the maximum deflection of manipulator arms. The maximum deflection of manipulator arms is not allowed to exceed a specific value δ_s. The goal of static discipline is to keep the weight of the manipulator close to a specific value w_s. The optimization computational time for static analysis is set to be 3.

$P_1 - Static$

Given

δ_s, l_s

Find

$r_1, r_2, r_3, l_1, l_2, l_3, d_1^+, d_1^-$

Satisfy

$$\delta = 100 \cdot 9.8[\frac{l_1^3}{3EI_1} + \frac{(l_2 + l_3)l_1^2}{2EI_1} + \frac{l_2^3}{3EI_2} + \frac{l_3 l_2^2}{2EI_2} + \frac{l_3^3}{3EI_3}]$$

$\delta < \delta_s$

$w = 2700\pi(r_1^2 l_1 + r_2^2 l_2 + r_3^2 l_3)$

$w - w_s + d_1^- - d_1^+ = 0$

$l_1 + l_2 + l_3 = l_s$

Minimize

$Z_1(d_1^-, d_1^+)$

(a) Activity Template of Static Discipline

$P_2 - Cost$

Given

$c_s, k_c, k_\tau, \delta_s, \tau_{s1}, \tau_{s2}, \tau_{s3}$

Find

$l_1, l_2, l_3, r_1, r_2, r_3, d_2^+, d_2^-$

Satisfy

$\tau_1 = 100 \cdot 9.8(l_1 + l_2 + l_3)$

$\tau_2 = 100 \cdot 9.8(l_2 + l_3), \tau_3 = 100 \cdot 9.8 l_3$

$\tau_1 < \tau_{s1}, \tau_2 < \tau_{s2}, \tau_3 < \tau_{s3}$

$$\delta = 100 \cdot 9.8[\frac{l_1^3}{3EI_1} + \frac{(l_2 + l_3)l_1^2}{2EI_1} + \frac{l_2^3}{3EI_2} + \frac{l_3 l_2^2}{2EI_2} + \frac{l_3^3}{3EI_3}], \delta < \delta_s$$

$c = k_c \pi(r_1^2 l_1 + r_2^2 l_2 + r_3^2 l_3) + k_\tau(\tau_1 + \tau_2 +$

$c - c_s + d_2^- - d_2^+ = 0$

$l_1 + l_2 + l_3 = l_s$

Minimize

$Z_2(d_2^-, d_2^+)$

(b) Activity Template of Cost Discipline

$P_3 - Geometry$

Given

$\tau_{s1}, \tau_{s2}, \tau_{s3}, p_s, l_s$

Find

$l_1, l_2, l_3, \theta_1, \theta_2, \theta_3, d_3^+, d_3^-$

Satisfy

$\tau_1 = 100 \cdot 9.8(l_1 + l_2 + l_3), \tau_2 = 100 \cdot 9.8(l_2 + l_3), \tau_3 = 100 \cdot 9.8 l_3$

$\tau_1 < \tau_{s1}, \tau_2 < \tau_{s2}, \tau_3 < \tau_{s3}$

$$p = 0.5 l_1 \cos\theta_1 (3 - l_2 \sin\theta_2 - l_3 \sin\theta_3) + 0.5 l_2 \cos\theta_2 (2 l_1 \sin\theta_1 + l_2 \sin\theta_2) + 0.5 l_3 \cos\theta_3 (2 l_1 \sin\theta_1 + 2 l_2 \sin\theta_2 + l_3 \sin\theta_3)$$

$p - p_s + d_3^- - d_3^+ = 0$

$l_1 \cos\theta_1 + l_2 \cos\theta_2 + l_3 \cos\theta_3 = 5$

$l_1 \sin\theta_1 + l_2 \sin\theta_2 + l_3 \sin\theta_3 = 3$

$l_1 + l_2 + l_3 = l_s$

Minimize

$Z_3(d_3^-, d_3^+)$

(c) Activity Template of Dynamic Discipline

Figure 7.2: Design Activity Template

For the geometry discipline, the constraint is the maximum torques on each joint. These joint torques should be kept below allowable limits. The goal of geometry discipline is to maximum the clearance space under manipulator arms. The maximum clearance space is expected to be close to the specific value v_s. It has been set that the optimization computational time for geometry discipline is 2.

The last discipline is cost. The goal of cost discipline is to reduce the manipulator cost. In this dissertation, the cost is assumed to be proportional to the volume of the manipulator arms and sum of the torques on each joint. The maximum cost should be close to the specific value c_s. It is set that the optimization computational time for cost discipline is 1.

7.1.2 MDD Representation

From the discussion in Chapter 5, different design collaboration strategies may be applied to form different design process organizations. Using the approach introduced in Chapter 5, it is possible to generate numerous organizations of design process. These organizations are described using MDD graphs and through evaluations the performances of each MDD graph are estimated and a suitable MDD graph which represents a design process organization satisfying with most of engineering requirements is selected based on performance evaluations.

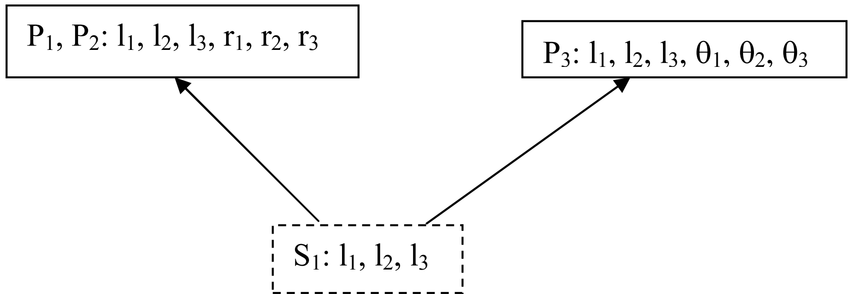


Figure 7.3: Maximal shared design variables

Starting from the analysis of activity dependency, the relationships of three design activities in manipulator example are decided by the shared variables that are used. In this example, design activities in static and cost disciplines need to use the design variable r_1, r_2, r_3, l_1, l_2, l_3. And design activity in geometry discipline use the design variables r_1, r_2, r_3, θ_1, θ_2, θ_3. Based on the information of design variables in each design activity, the maximum shared design variables for all three design activities can be derived (see Figure 7.3). The information of maximum shared design variables is helpful to find proper groupings for design activities. And each grouping of design activity leads to a possible way to organize the design process. In the manipulator example, because l1, l2 and l3 are shared by all three design activities, according to the approach in chapter 5, three alternative groupings are generated. However all of three groupings are the same. There is only one grouping available (see Figure 7.4) and in this grouping, all three design activities are solved together.

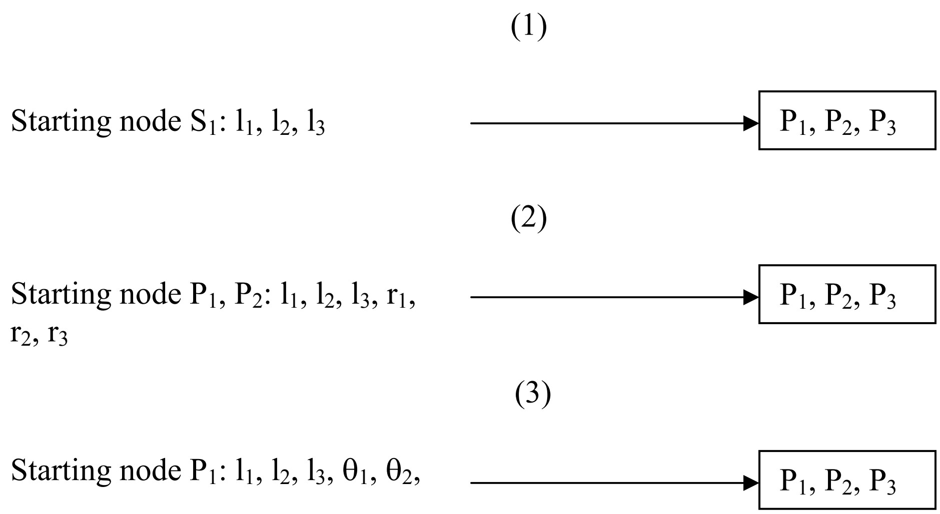
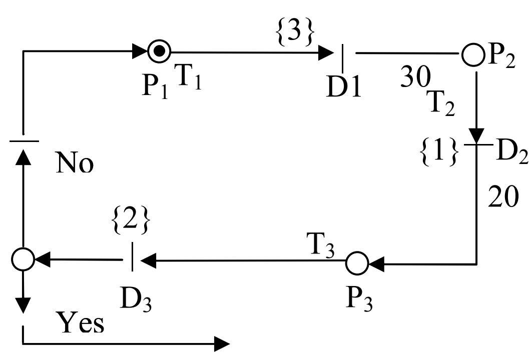


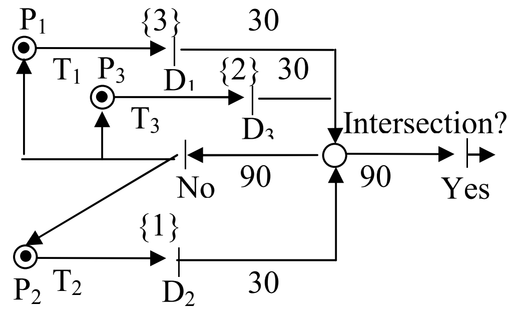
Figure 7.4: Available grouping

Based on grouping information, by choosing Pareto, Nash and Leader/following design collaboration strategies, three different MDD graphs can be generated in the manipulator example. Following this MDD graph generation approach, three MDD graphs (see Figure 7.5-7.7) are developed to solve the three coupled design activities.



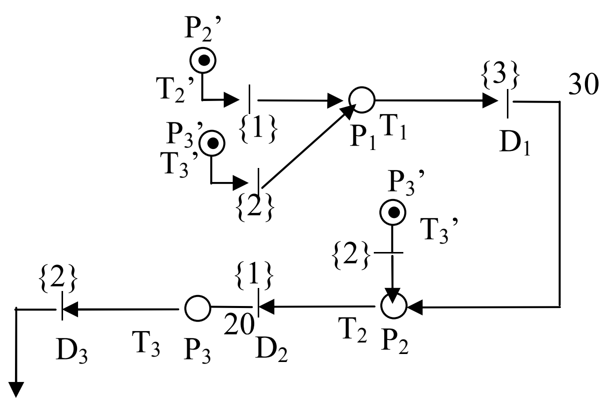
Tokens:
T_1=(Mv_1: l_1, r_1 ; Sv_1: l_2, l_3, r_2, r_3; $RS_1(l_1, l_2, l_3, r_1, r_2, r_3)$)
T_2=(Mv_2: r_2, r_3 ; Sv_2: l_2, l_3; $RS_2(l_1, l_2, l_3, r_1, r_2, r_3)$)
T_3=(Mv_3: l_2, l_3, θ_1, θ_2, θ_3; Sv_2; $RS_2(l_1, l_2, l_3, \theta_1, \theta_2, \theta_3)$)

Figure 7.5: Pareto MDD graph



Tokens:
T_1=(Mv_1: l_1, l_2 ; Sv_1: l_3, r_1, r_2, r_3;)
T_2=(Mv_2: r_1, r_2, r_3 ; Sv_2: l_1, l_2, l_3; RS_2(l_1, l_2, l_3, r_1, r_2, r_3))
T_3=(Mv_3: l_3, θ_1, θ_2, θ_3; Sv_2: l_1, l_2; RS_2(l_1, l_2, l_3, θ_1, θ_2, θ_3))

Figure 7.6: Nash MDD graph



Tokens:
T_1=(Mv_1: l_1, r_1 ; Sv_1: l_2, l_3, r_2, r_3)
T_2=(Mv_2: r_2, r_3 ; Sv_2: l_2, l_3)
T_3=(Mv_3: l_2, l_3, θ_1, θ_2, θ_3; Sv_2)
T_2'=RRS_2(l_1, l_2, l_3, r_1) T_3'=RRS_3(l_1)

Figure 7.7: Leader/follower MDD graph

The evaluations of the above three MDD graphs are given in Table 7.1. From the results, Nash MDD graph has better design quality and shorter design time. However the implementation of Nash MDD graph requires more design resources than other two MDD graphs, Pareto and leader/follower. The Pareto and leader/follower MDD graphs have similar design qualities, overall design times and resource consumptions. Based the performances of three MDD graphs, selection of MDD graph can be made. In our example, if design quality and design time have higher priorities, Nash MDD graph is selected for the implementation of design process. In different situations, priorities of engineers may vary and the final selection of MDD graphs also can be different to meet engineering requirements.

Evaluation	Quality of design	Overall design time	Concurrency ratio	Design resources
Pareto	0.55	112	0	51
Nash	0.7	90	0.5	120
Leader/follower	0.55	114	0.03	54

Table 7.1: Evaluations of MDD graphs

7.1.3 Implementation of Product Development

The last step of the example is to implement the product development according to the MDD graph. From this step, various design tools and design systems are involved to help achieve design collaboration among engineers in a distributed environment. In this manipulator example, the Nash MDD graph is selected by engineers and the design activities in the MDD graph are treated as design tasks assigned to different engineers.

Based on design tasks and their sequence, a schedule of all tasks and responsible engineers are inputted into process management system. The process management system automatically assigns tasks to the corresponding engineers to generally progress product development. In the manipulator example, three tasks are assigned to three engineers at the initial time. After product development starts, all three tasks are performed simultaneously and each task generates a set of design solutions. The intersection of three sets of design solutions is the final design for manipulator.

Using the approach introduced in chapter 4 about design decision making, engineers in different design discipline work separately. By applying Nash design collaboration strategy, engineers do not need to exchange the design information to let others to understand their design works.

The results of each design tasks are calculated using optimization software VisualDoc. In this example, the elastic module of the material is set as 210G. The overall length of manipulator arms l_s is set as 6m. The weight of manipulator is set as 600kg and the maximum deflection is set as 6mm. The results of design and analysis in static discipline are listed in Table 7.2. Figure 7.8 shows the user interface of VisualDoc for optimization.

l_1	l_2	l_3	r_1	r_2	r_3	Objective
2.1	2.1	1.782	0.07212	0.06617	0.05458	626.7
2.1	2.0	1.902	0.07211	0.06590	0.05585	629.0
2.1	1.9	1.987	0.07155	0.06463	0.05846	628.7
2.0	2.1	1.884	0.07181	0.06655	0.05516	624.6
2.0	2.0	2.017	0.07212	0.06655	0.05684	633.4
2.0	1.9	2.084	0.07173	0.06551	0.05754	625.7
1.9	2.1	1.988	0.7179	0.06645	0.05636	626.3
1.9	2.0	2.084	0.07102	0.06625	0.05777	625.0
1.9	1.9	2.180	0.07086	0.06581	0.05870	624.1

Table 7.2: Results of design activity in static discipline

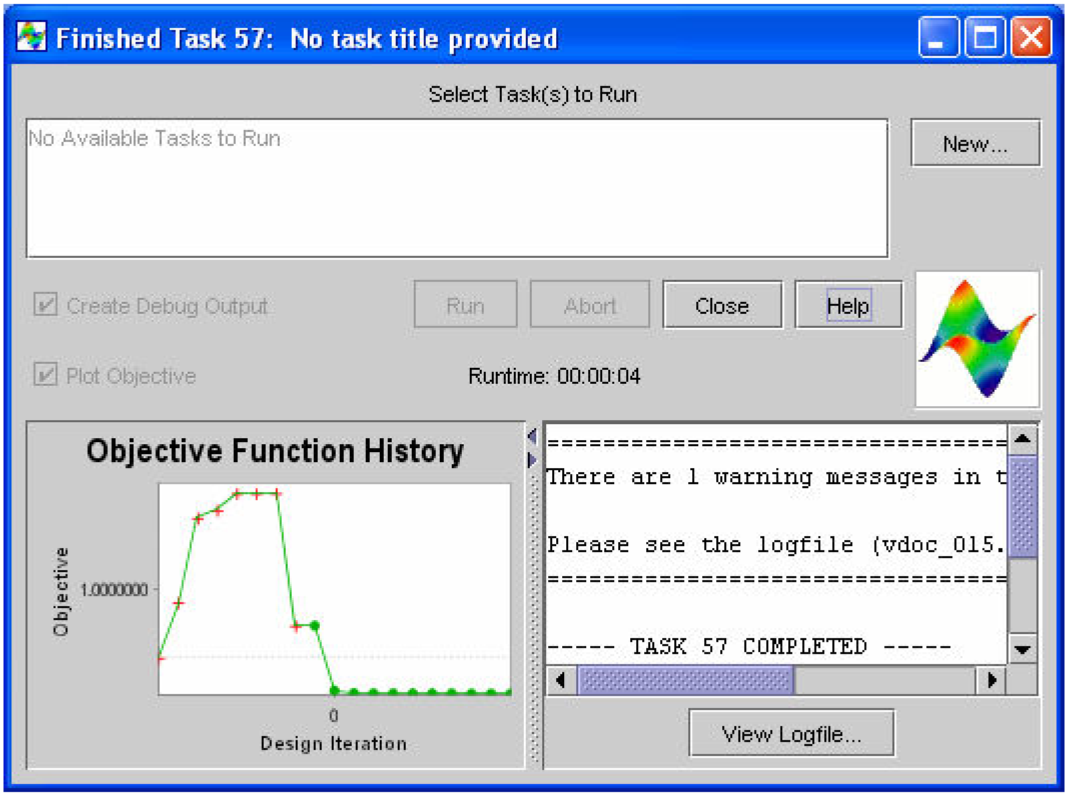


Figure 7.8: VisualDoc interface of optimization in static discipline

The results of design activity in geometry discipline are given in Table 7.3. The specified limits of torque in three joints are set as 6000Nm, 4000Nm and 2100 Nm respectively. Figure 7.9 shows the VisualDoc interface for optimization in geometry discipline.

l_3	l_1	l_2	θ_1	θ_2	θ_3	Objective
1.9	2.068	2.034	0.8417	0.5211	0.2389	9.799
2.0	2.013	1.991	0.8555	0.5058	0.2553	9.802
2.1	2.112	1.778	0.9210	0.5377	0.1559	10.09

Table 7.3: Results of design activity in geometry discipline

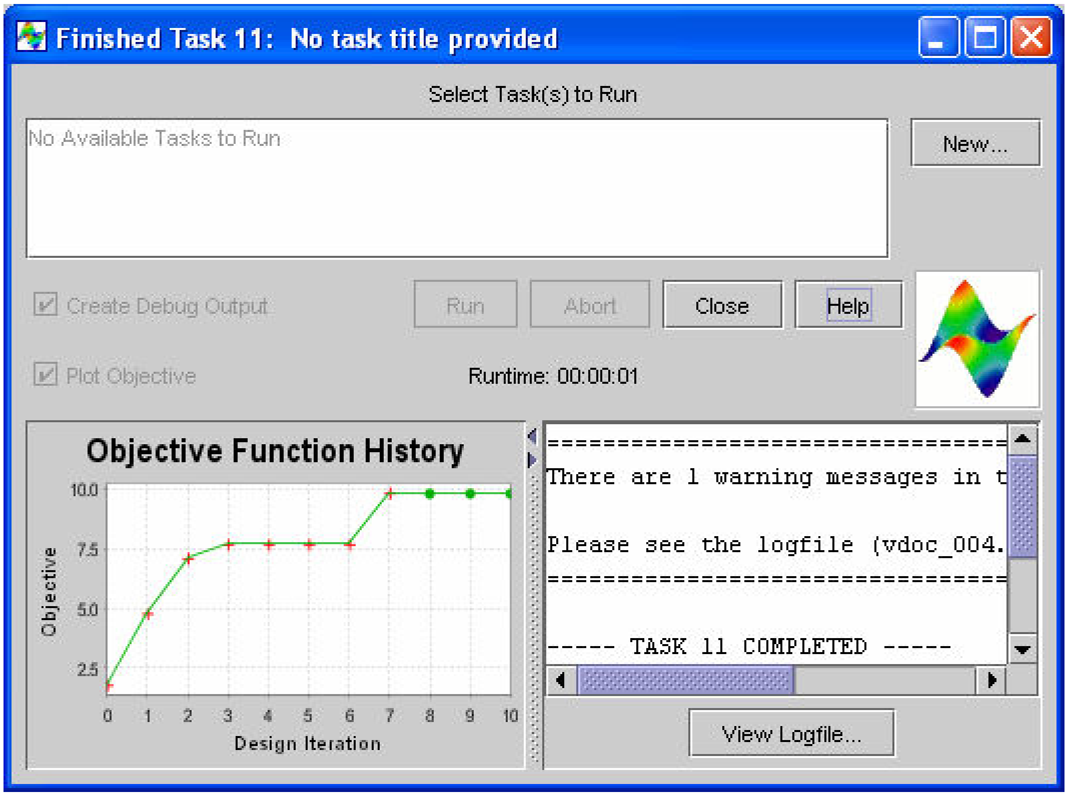


Figure 7.9: VisualDoc interface of optimization in geometry discipline

From the comparison of the design results in above two disciplines, it can be found that the second rows in Table 7.2 and the first row in Table 7.3 is a possible solution intersection. The values of shared design variables are approximately equal in the two disciplines. In cost discipline, the above design solution intersection is also an acceptable and satisfying design solution. In Table 7.4, when the two cost factors k_c and k_τ are specified as 1 and 0.0001, the result is calculated and shown in Table 7.4.

r_1	r_2	r_3	l_1	l_2	l_3	Objective
0. 07211	0. 06590	0. 05585	2.1	2.0	1.9	1.236
0.07155	0.06463	0.05846	2.0	1.9	2.0	1.244
0.07181	0.06655	0.05516	2.0	2.1	1.9	1.537

Table 7.4: Results of design activity in cost discipline

According to the design results in three disciplines, the Nash solution is the intersection of three design solutions generated from three design activities. The final design solution of the manipulator design is selected as l_1=2.1, l_2=2.0, l_3=1.9, r_1=0. 07211, r_2=0. 06590 and r_3=0. 05585.

7.2 Example of Reverse Engineering Torque Tube Design

The example in Section 7.1 shows how the design approaches introduced in Chapters 4 and 5 are applied to perform distributed product development. The focus of the example is to provide an understanding of distributed mechanical design from the aspect of design decision making. Besides design decision making, there are some other design activities which are also important for product development such as modeling, analysis and simulation. Design support systems are required for achieving successful distributed product development with various design activities taken by different design disciplines.

A reverse engineering example is provided in this section to demonstrate how design support systems can support distributed product development. The assumptions of this section are that engineers have finished the meta-design stage and a MDD graph is selected for the organization of design decision activities. As mentioned earlier the analyses work for this example was performed by Prof. Chang and his students [124].

A MDD graph only defines design decision activities. To further complete all design activity information, non design decision technical and social activities are involved in product development. All non design decision activities and design decision activities are the foundation for making a workflow that organizes design works. Many software products support workflow creation. In this example, the workflow management tool in PTC Windchill is applied to create the workflow (see Figure 7.10). In Figure 10, the FEA engineer and virtual manufacture engineer are organized in a way that they work simultaneously with complete design information sharing. Design conference is held to make engineers in different disciplines fully understand the design works in other disciplines. This kind of organization is a Pareto design collaboration strategy. Additional tasks in the workflow in Figure 7.10 includes non decision making technical activities such as design information uploading, report review, etc. and social activities such as conference, email notification, etc. the workflow makes it possible to schedule design tasks to engineering team members to actually complete product development. According to workflow, design tasks with task descriptions are assigned to engineers properly. Dependent tasks in the workflow are only launched when predecessor tasks are completed successfully without problems.

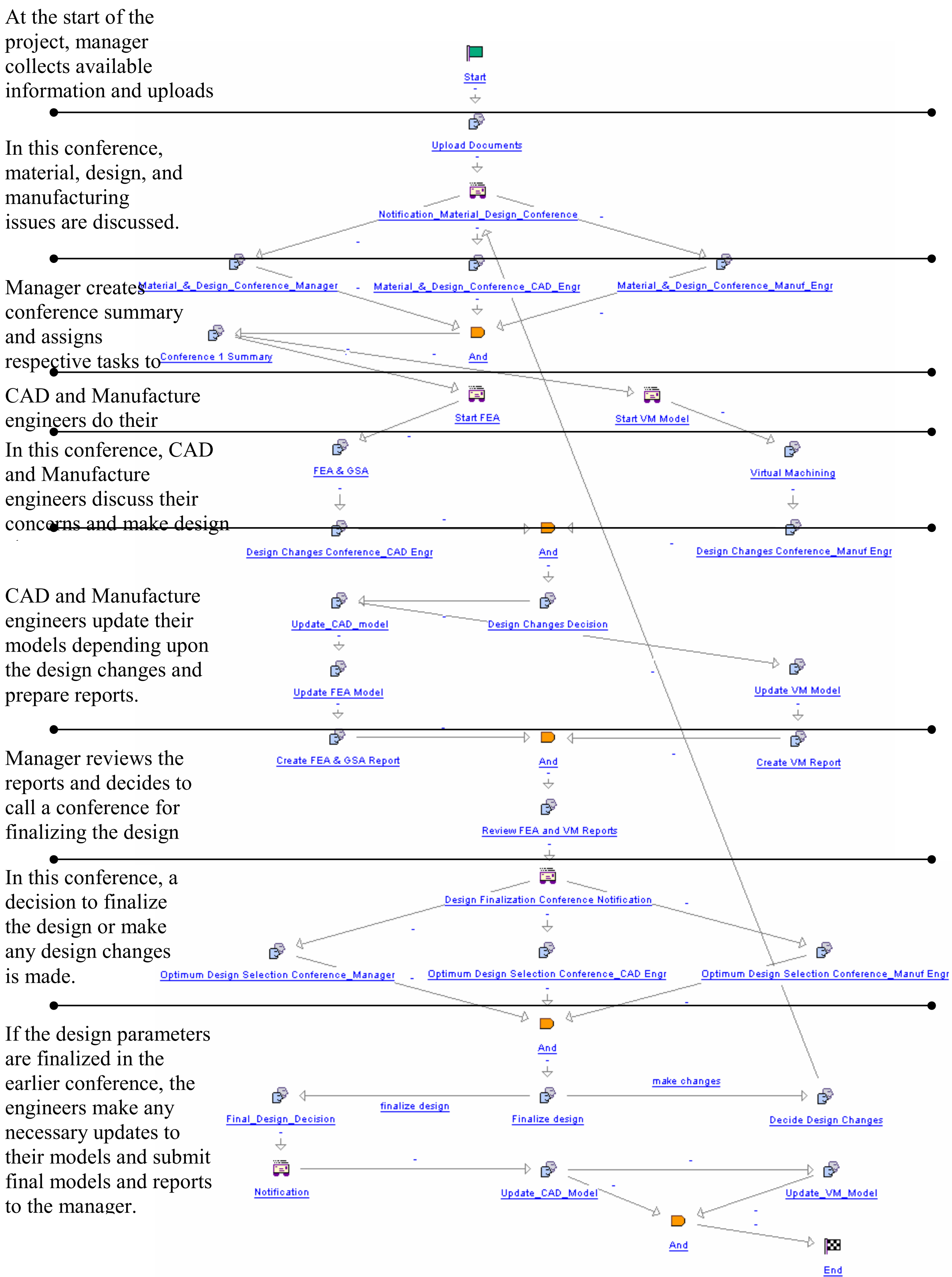


Figure 7.10: Workflow of decision based and non decision based tasks

To complete design tasks in the workflow, engineers in distributed locations need an integrated design system to support reverse engineering, re-engineering, and fast prototyping. The reverse engineering aims at not only reconstructing solid models from physical sample parts, but more importantly, constructing parametric solid models with geometric features and dimensions. Usually, the NURB (Non-Uniform Rational B-spline) surface models are sufficient for reverse engineering if not considering re-engineering. However, in order to support re-engineering, geometric features embedded in the NURB surface model must be recognized and properly parameterized.

The re-engineering focuses on incorporating fatigue and fracture computations as well as shape optimization for optimal or near-optimal component designs. Computer modeling and simulation tools, such as multibody dynamic simulations, finite element analysis (FEA), and fatigue and fracture prediction techniques have been employed to simulate the fatigue and fracture behavior of the failed parts. Based on the simulation results, material and part geometry can be optimized for required performance with a minimum cost (or minimum part weight in most cases). In the fast prototyping, the solid freeform fabrication (SFF) technology (also called Rapid Prototyping) is employed to fabricate physical prototypes of the re-engineered parts for design verification. At the same time, virtual machining and metal forming simulations will support manufacturing process planning and simulation before fabricating the functional prototype or embarking parts manufacturing. An integration framework has been developed using Windchill of

Parametric Technology Co. to embrace the tools and technology involved, support design collaboration, and facilitate information sharing and project management.

The presented Reverse, Re-engineering and Fast prototyping (RRF) processes involve using different techniques, technologies and software. To efficiently accomplish the design tasks in each RRF step, advanced computer based tools are required. These heterogeneous tools usually use different file format, work on different platform and thus difficult to be integrated in one design environment. In this research, the focus is not on converting file formats or interoperability of CAD/CAM/CAE software. Our integration concern is to select proper available commercial software and allow the built-in compatibility of the software to meet the integration needs in reverse engineering. Most reverse engineering solutions involve multidisciplinary design activities. Consequently, design collaboration is essential for a typical reverse engineering project to let multiple designers in different disciplines perform their roles. In the integration system, the design collaboration is based on two kinds of designers' interactions: asynchronous and synchronous. Asynchronous interactions involve email, notification, forums as well as sharing documents where the designer is not required to respond in real-time. During synchronous interactions, the designer is required to response at real-time. These synchronous interactions include white board, chat room, model viewer, video and audio communication and so on. To meet these requirements the integrated environment supports:

• Appropriate distribution of activities to members of the team;

• Tools that can support real-time collaboration among team members with engineering information;

• An environment that organizes and provides easy access to engineering and other information related to the project for the team;

• A knowledge base that includes information related to different reverse engineering processes, tools and techniques;

• A reverse engineering template that can be modified to support different reverse engineering processes and reduce the initial effort to setup products.

The RRF testbed is intended to provide an environment that is software independent and can support multiple geographically dispersed designers. This principle extends to all reverse engineering activities, data, and collaborative activities, as well as to the infrastructure design. The testbed is setup using simple client-server architecture. The Windchill and communication module is housed in the server and is connected to the Internet. Multiple clients (users) access product and reverse engineering information from the servers using a web browser environment. Some product management functions supported by the servers are: (1) managing the product data and model in a structure through which a designer can easily locate the product data; (2) keeping the data secure and restrict illegal operations through basic file access controls; (3) providing functions to manage the file operation privileges based on designers' roles in the team; and (4) supporting file status control to prevent the file inconsistency which may occur when two users modify the same file simultaneously. In order to support real-time collaboration, a web-based tool has been developed (see Figure 7.11). This collaborative tool supports text messaging, audio, video, sketching, and viewing of 3D models in real-time to facilitate activities required for meetings. To enhance collaboration among different members of the team, the collaborative 3D model viewer allows users to have real-time synchronous view of the model, add notes to the 3D model, and exchange text and audio information in real-time. Collaborative meetings, if needed, can be scheduled in an adhoc manner. When a meeting is scheduled, appropriate group members are sent an email that has the web-link to the collaborative tool and the scheduled meeting time. During the scheduled time all group members can log into the collaborative tool to discuss issues related to the project using the environment. Client interface providing tasks list and information, product structure, engineering data and booked rules.

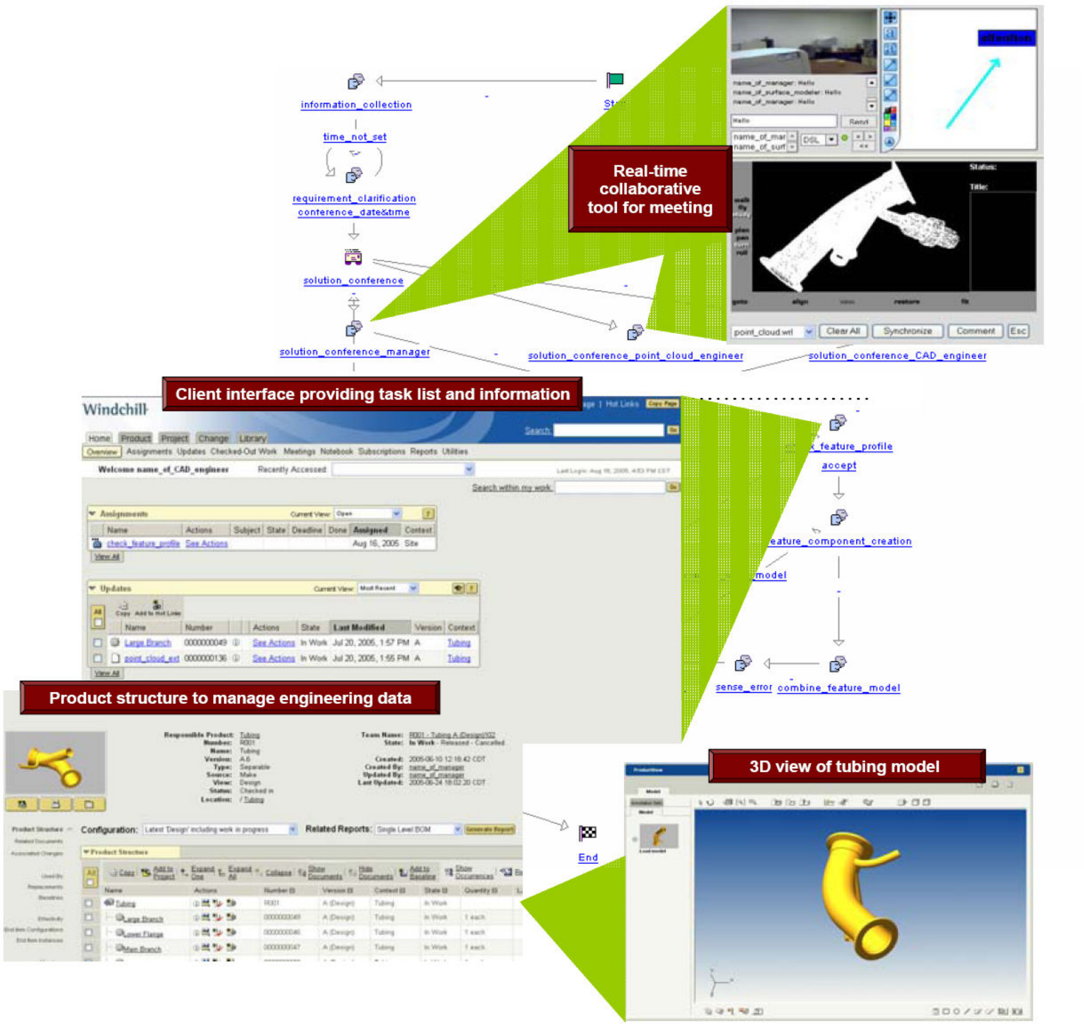


Figure 7.11: Reverse engineering integrated environment adapted from [120]

In order to evaluate the testbed, a case scenario was created. The reverse and re-engineering scenario highlights (1) a systematic reverse engineering approach, (2) an enhanced ability of team members collaboration, and (3) a customized Windchill product management system. The reverse engineering of the B-52 anti-icing tubing scenario involves an engineering team consisting of four members, who are geographically distributed: Manager, CAD Engineer, and two Point Cloud Engineers. A template with a flow of activities (see Figure 7.10), along with appropriate instructions, has been setup in the Windchill environment. This template is the start point for the manager to initiate a reverse engineering project. The initial steps for the manager involve gathering information, design constraints and point-cloud information for the product. Once the information has been gathered, the manager creates the team and calls a meeting in the integration framework using the real-time collaborative tools (Figure 7.11) to discuss details of the project. After the meeting appropriate reverse engineering process can be selected and modified according to the requirement and need of the project. The integration framework then supports accomplishing these tasks by appropriate users. Information and instruction on how to complete the different tasks are also available to the users from the environment. Information created from each activity is uploaded in the environment for other members of the team to view, access, evaluate and use. These data are organized in a set of defined folders that follow the product structure to reduce the effort of finding the files (see Figure 7.11). The progress of the project can be monitored by any member of the team at any given time. After each task is completed the environment sends appropriate notification to relevant team members to proceed to the next upload and download.

7.3 Summary

Chapter 7 presents two examples of using the design approaches and system introduced in this dissertation to develop products. The first example is focused on the collaborative decision making and design process modeling. It shows how these approaches are used to solve a manipulator design. The second example is focused on the design system and tools. It provides a design scenario of a group of engineers working in distributed environment and accomplishing a reverse engineering project. Through the second example, more detailed information about system level mechanical design is introduced. Workflow assignment, CAD data exchange, collaborative design tools, etc are all applied to build a testbed for distributed design.

CHAPTER 8
CONCLUSION

8.1 Research Summary

The focus of this dissertation is on characterization and development of a distributed mechanical design framework. Based on the framework, various important elements for distributed mechanical design are presented. Among them, design decision is a key element to achieve successful collaborative design. The approaches presented in this dissertation guide engineers to search for proper design solutions that satisfy requirements of multiple disciplines. Treated as the key element, design decision plays an important role in design process organization. Models of the design process are developed in this dissertation to describe multiple design activities, especially design decisions and their dependent relationship.

The engineering requirements of the design system are discussed to gain a better understanding of the distributed mechanical framework, which helps to better understand the requirements for an ideal distributed mechanical design. Besides design decision, there are other elements in the framework which can only be achieved by using design support systems. The system architecture, components and functions of the implemented design system have also been investigated. The models that are applied to handle synchronization and communication issues of the Group Design Tools have been discussed. Several group design tools have been developed and integrated in the design system to facilitate collaboration among a team of distributed designers. This research provides an opportunity to realize distributed product development from the starting point of organizing a distributed design process to the ending point of obtaining all product development data. The goal is achieved by the approaches and systems developed in this dissertation that enhance the collaboration among engineers in multiple disciplines and geographically distributed.

The outcomes of the research are the approaches that support distributed design decision making, a design system that supports various group design activities in each design stage, and engineering process and data management based on the requirements of collaboration between designers in different design teams.

8.2 Answers to Research Questions

The research works in this dissertation attempt to answer some fundamental questions that are mentioned in Section 1.3. The research work in this dissertation starts from answering the questions:

(v) What are required to accomplish a distributed mechanical design?

From the discussion in Chapter 3, several elements are essential for a successful distribute collaborative design. These elements include technical and social activities, communication, synchronization, coordination, product data management, process organization, resource utilization, knowledge representation, information sharing, etc.

(vi) How can distributed engineering teams make design decisions separately and achieve proper collaboration?

Design decision making is an important type of technical activity. In this dissertation, Chapter 4 attempts to address the decision making issue in a distributed environment, where engineers may not be able to meet with each other. The answer to this question includes three parts. First decision making activity should be represented in a format so that design information can be shared among relevant engineers. c-DSP has been used as an template to achieve this requirement. Second game protocols are used to construct individual design activities into a design game. Third in order to apply leader/follower protocol, design freedom needs to be maintained and the modified robust design formulation in Section 4.3 can be used to find proper solution range.

(vii) How can engineers organize a distributed design process?

A design process can be treated as a game construct of individual design activities based on three game protocols. The research in Chapter 5 shows that it is possible to represent, generate, and evaluate different combinations of design activities. The representation and evaluation information help engineers find a suitable combination to organize design activities.

(viii) What group design system is needed to facilitate engineers in a distributed design process?

Based on the discussion in Chapter 6, group design system should be equipped with various functions to support real time group design activities and provide a platform to manage product data, design process, etc. Many special designed real time design tools need to be implemented. Compare with traditional CAD tools, these real time tools can synchronize the data in distributed locations.

8.3 Research Contributions

Product development is a complex process that needs the collaboration among individuals or teams in a global company. This dissertation presents the distributed mechanical design framework which is an overview of the collaborative distributed mechanical. The framework explicitly addresses various types of elements for a successful distributed mechanical design, classifies the design activities into social and technical aspects and defines the interactions during design collaboration into three major types: synchronization, communication, and coordination. From the survey of researches, the types of design interactions are not explicitly mentioned in some other research.

Besides an overall of distributed mechanical design, the researcher presents the detailed approaches for engineers to make design decision separately within their own discipline and at the same time consider the needs from other disciplines. Game theory is used to classify the interactions between engineers. Different from some other game based design research [54, 57], in this dissertation a new approach for managing design freedom is presented. This is an essential step to achieve successful mechanical design using the leader/follower strategy.

To provide an accurate method for design process modeling, in this dissertation a Petri-net model has been introduced and presented. Compared with other research, the Petri-net design process model explicitly describes the relationship of multiple design activities, which brings more information for engineers to understand a design process. The research in this dissertation is the first time to use Petri-net to model design decision process. Based on the Petri-net model, evaluation, and analysis are available, which can facilitate investigation of characteristics of various design process alternatives. In this dissertation, the measurement of quality of design and some other evaluations are mentioned to help engineers understand how good their design processes are so that they can select the right design process for their product development.

In the area of the GDS tool development, the needs for design tools in different design stages are addressed, by proposing synchronous collaboration among designers, especially conceptual design stage. These tools are not only developed according to engineers' needs, but are also equipped with synchronization mechanism to ensure real-time collaboration among engineers. Unlike other stand-alone CAD software, the developed design tools facilitate multi-users to perform the specific design activities in conceptual design stage.

Although design tools are important for engineers to perform their tasks, a management system is also important for engineers in different locations to work together. In this dissertation, the design support system includes process management, data management and user management, which are prerequisites for a group of engineers in different locations to get assignments, share data, and play different roles in a design process.

8.4 Research Limitations

In this dissertation, there exist many research limitations that may influence the potential applications of the developed approaches and design system. In Chapter 4, the decision making approach is developed based on continuous variables, whereas in many cases, discrete variables are needed. In Chapter 5 the approach for modeling design process is provided. However only design decision activities are considered in the design process model. Other design activities such as creating geometric models are not able to be represented using current approach. As to the design system and tools, still many essential functions need to be provided to engineers. For example, there is not a static analysis tools that supports a group of distributed engineers to work together to set up analysis conditions and perform analysis. Further research and development is required to improve current design system and tools.

References

[1] Nahm, Y. E., Ishikawa, H. (2004) “Integrated Product and Process Modeling for Collaborative Design Environment”, *CERA Journal*, 12(1), March, pp. 5-23

[2] Jin, Y., Lu, S. C-Y., 1998, “Toward a Better Understanding of Engineering Design Models”, Proceedings of Universal Design Theory Workshop, May 12-138

[3] Lu, S. C-Y., Cai, J., 2001, “A Collaborative Design Process Model in he Sociotechnical Engineering Design Framework”, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 15, pp. 3-20

[4] Fujita, K., and Akagi, S., 1998, “Agent-based Distributed Design System Architecture for Basic Ship Design”, Proceedings of ASME Design Engineering Technical, Design Automation Conference, Sacramento, CA, DETC1997/DAC-21106

[5] Sobieszczanski-Sobieski, J., Haftka, R. T., 1997, “Multidisciplinary Aerospace Design Optimization: Survey of Rcent Developments”, *Structural Optimization*, 14(1), pp. 1-23

[6] Kim, J., King, J., 2004, “Managing Knowledge Work: Specialization and Collaboration of Engineering Problem-solving”, *Journal of Knowledge Management*, Vol. 8 Num. 2, pp. 53-63

[7] Berkes, U. L., 1990, “Efficient Optimization of Aircraft Structures with a Large Number of Design Variables”, *Journal of Aircraft*, 27(12), pp. 1073-1078

[8] Wynn, D., Eckert, C., Clarkson, P. J., 2005, “Abstracting Complexity for Design Planning”, GIST Technical Report G2005-1, Trumpington St, Cambridge, UK.

[9] MDO pages, http://www.soton.ac.uk/~pbn/MDO/

[10] AIAA Technical Committee on Multidisciplinary Design Optimization (MDO) White Paper on Current State of the Art, http://endo.sandia.gov/AIAA_MDOTC/sponsored/aiaa_paper.html

[11] Sobiesezczanski-Sobieski, J., 1988, "Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems", Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, NASA CP 3031, Part I, pp. 51-78, ed. J-F. M. Barthelemy, Hampton, VA.

[12] Bloebaum, C. L., Hajela, P., and Sobieski, J., "Non-Hierarchic System Decomposition in Structural Optimization", *Engineering Optimization*, Vol. 19, pp. 171-186, 1992.

[13] Renaud, J. E., and Tappeta, R. V., 1997, "Multiobjective Collaborative Optimization", *ASME Journal of Mechanical Design*, Vol. 119, No. 3, pp. 403-411

[14] Balling, R. J., and Sobieski, 1994, J., "An Algorithm for Solving the System-Level Problem in Multilevel Optimization", 5th AIAA/USAF/NASA/ISSMO Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, Panama City, FL, Part II, pp. 794-809.

[15] Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I., "Multidisciplinary Optimization Methods for Aircraft Preliminary Design", Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, September 7-9, Panama City, FL, Part I, pp. 697-707, AIAA 94-4325, 1994.

[16] Braun, R.D., Kroo, I.M., and Moore, A.A., 1996, "Use of the Collaborative Optimization Architecture for Launch Vehicle Design", 6th AIAA/USAF/NASA/ISSMO

Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, Sept. 4-6, Part I, pp. 306-318, AIAA 96-4018.

[17] Braun, R. D., Moore, A. A., and Kroo, I. M., 1997, “Collaborative Approach to Launch Vehicle Design”, *Journal of Spacecraft and Rockets*, 34(4), pp. 478-486

[18] Kroo, I. M., Altus, S., Grage, p., Braun, R., and Sobieski, I., 1994, “Multidisciplinary Optimization Methods for Aircraft preliminary Design”, Fifth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, Panama City, FL, AIAA 94-4325

[19] The PACE Project: A Practical Approach to Concurrent Engineering, http://www.biba.uni-bremen.de/projects/pace/

[20] Vincent, T. L., 1983, “Game Theory as a Design Tool”, *ASME Journal of Mechanisms*, *Transmissions, and Automation in Design*, Vol. 105, pp. 165-170, June 1983.

[21] Lewis, K., 1996, “An Algorithm for Integrated Subsystem Embodiment and System Synthesis”, Ph.D. Dissertation, G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, August 1996.

[22] Wang, J. F., Periaux, J., 2001, “Multi-Point Optimization using GAS and Nash/Stackelberg Games for High Lift Multi-airfoil Design in Aerodynamics”, *IEEE*, 0-7803-6657-3/0

[23] Ullman, D., 1988, “A Model of the Mechanical Design Process based on Empirical Data”, *AI EDAM*, 2(1), pp. 33-52

[24] Lockledge, J. C., and Salustri, F. A., 1999, “Defining the Engine Design Process”, *Journal of Engineering Design*, 10(2), pp. 109-124

[25] Chen, L., Li, S., 2005, "Analysis of Decomposability and Complexity for Design Problems in the Context of Decomposition", *Journal of Mechanical Design*, Vol. 127, pp. 557

[26] Guenov, M. D., 2002, "Complexity and Cost Effectiveness Measures for Systems Design, Manufacturing Complexity Network Conference", April 9-10 2002, Downing College, Cambridge, UK

[27] Loch, C., Mihm, J., Huchzermeier, A., 2003, "Concurrent Engineering and Design Oscillations in Complex Engineering Projects", *Concurrent Engineering: Research and Applications*, Vol. 11, Num. 3, pp. 187-197

[28] Ford, D. N., Sterman, J. D., 2003, "Overcoming the 90% Syndrome: Iteration Management in Concurrent Development Projects", *Concurrent Engineering: Reseaerch and Application*, Vol. 11, Num. 3, pp. 177-186

[29] Ansys solutions, http://www.ansyssolutions.com/ansys_solutions_summer03.pdf

[30] Bras, B. A., Mistree, F., 1991, "Designing Design Processes in Decision Based Concurrent Engineering", SAE Transactions, *Journal of Materials & Manufacturing*, Vol. 100, pp. 451-458

[31] Mistree, F., Muster, D., Shupe, J. A., and Allen, J. K., 1989, "A Decision based Perspective for the Design of Methods for Systems Design", Recent Experience in Multidisciplinary Analysis and Optimization, Hampton, Virginia

[32] Mistree, F., Muster, D., Shupe, J. A., Allen, J. K., 1989, "A Decision based Perspective for the Design of Methods for Systems Design", Book Chapter in: Recent Experience in Multidisciplinary Analysis and Optimization, Ed. Hampton, Virginia, NASA CP 3031

[33] Bras, B., Smith, W. F., Mistree, F., 1990, "The Development of a Design Guidance System for the Early Stages of Design, Book Chapter in: CFD and CAD in Ship Design", Ed. Oortmerssen, G. V., pp. 221-231, Wageningen, The Netherlands, Elsevier Science Publishers B. V.

[34] Mistree, F., Marinopoulos, S., Jackson, D., Shupe, J. A., 1988, "The Design of Aircraft Using the Decision Support Problem Technique", Systems Design Laboratory, Department of Operation Research, University of Houston

[35] Mistree, F., Kamal, S. Z., and Bras, B. A., 1989, "Dsides: Decision Support in the Design of Engineering Systems", Systems Design Laboratory Report, University of Houston, Houston, Texas

[36] Mistree, F., Muster, D., Woods, R. T., 1990, "Conceptual Models for Decision based Concurrent Engineering Design for the Life Cycle", Proceedings of the Second National Symposium on Concurrent Engineering, pp. 443-467, Morgantown, West Virginia

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

[38] Mistree, F., Allen, J. K., Attia, F., 1993, "Designing at a High Level of Abstraction", *Behavioral Science*, Vol. 38, pp. 124-137

[39] Muster, d., Mistree, F., 1988, "The Decision Support Problem Technique in Engineering Design", *International Journal of Applied Engineering Education*, Vol. 4, Num. 1, pp. 23-33

[40] Shupe, J. A., Muster, D., Allen, J. K., Mistree, F., 1988, "Decision based Design: Some Concepts and Research Issues", Book Chapter in: Expert Systems, Strategies and

Solutions in Manufacturing Design and Planning, Ed. Kusiak, A., Chapter 1, pp. 3-37, Dearborn, Michigan, Society of Manufacturing Engineers

[41] Shupe, J. A., 1988, "Decision based Design: Taxonomy and Implementation", Ph.D. Dissertation, Department of Mechanical Engineering, University of Houston, Houston, Texas

[42] Wassenaar, H. J., Chen, W., Cheng, J., and Sudjianto, A. 2003, "Enhancing Discreet Choice Demand Modeling for Decision-based Design", Proceedings of ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, September 2-6, 2003, Chicago, Illinois.

[43] Mistree, F., Smith, W. F., Kamal, S. Z., Bras, B. A., 1991, "Designing Decisions: Axioms, Models and Marine Applications", Fourth International Marine Systems Design Conference, pp. 1-24, Kobe, Japan

[44] Mistree, F., Bras, B., 1991, "Concurent Engineering: A Decision based Approach", Systems Design Laboratory, Department of Operation Research, University of Houston, Rept. Report No. SDL/REP.9/91

[45] Mistree, F., Hughes, O. F., Bras, B. A., 1993, "The Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm", Structural Optimization: Status and Promise, AIAA, pp. 249-289, Washington, D. C.

[46] Koch, P. N., 1997, "Hierarchical Modeling and Robust Synthesis for the Preliminary Design of Large Scale Complex Systems", Ph. D. Dissertation, School of Mechanical Engineering, Georgia Institute of Technology, Atlanta

[47] Peplinski, J., Koch, P. N., Allen, J. K., Mistree, F., 1996, "Design using Available Assets: A Paradigm Shift in Design for Manufacture", *Concurrent Engineering: Research and Applications*, Vol. 4, Num. 4, pp. 317-332

[48] Ignizio, J. P., 1985, "Multiobjective Mathematical Programming via the Multiplex Model and Algorithm", *European Journal of Operational Research*, Vol. 22, pp. 338-346

[49] Simon, H. A., 1996, The Sciences of the Artificial, 3rd Edition, The MIT Press, Cambridge, MA

[50] Xiao, A., Allen, J. K., Rosen, D., Mistree, F., 2000, "A Method to Design Product Architecture in a Distributed Product Realizaiton Environment", IEEE 9th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2000), pp. 124-130, Gaithersburg, Maryland

[51] Rao, S. S., 1987, "Game Theory Approach for Multiobjective Structural Optimization", *Computers & Structures*, Vol. 25, Num. 1, pp. 119-127

[52] Periaux, J., Chen, H. Q., Mantel, B., Sefrioui, M., Sui, H. T., 2001, "Combining Game Theory and Genetic Algorithms with Application to Ddm-Nozzle Optimization Problems", *Finite Elements in Analysis and Design*, vol. 37, Num. 5, pp. 417-429

[53] Badhrinath, K., Rao, J. R. J. 1996, "Modeling for Concurrent Design using Game Theory Formulations", *Concurrent Engineering Research and Application*, Vol. 4, Num. 4., pp. 389-399

[54] Lewis, L., Mistree, F., 1998, "Collaborative, Sequential, and Isolated Decisions in Design", *ASME Journal of Mechanical Design*, Vol. 120, Num. 4, pp. 643-652

[55] Fudenberg, D., Tirole, J. 1998, "Game Theory", The MIT Press, Cambridge, Massachusetts

[56] Lewis, K., and Mistree, F., 1997, "Modeling the Interactions in Multidisciplinary Design: A Game-Theoretic Approach", *AIAA Journal of Aircraft*, Vol. 35, No. 8, pp. 1387-1392, 1997.

[57] Chen, W., and Lewis, K. 1999, "A Robust Design Approach for Achieving Flexibility in Multidisciplinary Design", *AIAA Journal*, 7(8), 982-989.

[58] Parkinson, A., Sorenson, C., Pourhassan, N, 1993, "A General Approach for Robust Optimal Design", *ASME Journal of Mechanical Design*, Vol. 115, No. 1, pp. 74-80.

[59] Phadke, M.S., 1989, "Quality Engineering using Robust Design", Prentice Hall, Englewood Cliffs, New Jersey

[60] Yu, J., and Ishii, K., 1994, "Robust Design by Matching the Design with Manufacturing Variation Patterns", ASME Design Automation Conference, ed. B. Gilmore, September 1994, Minneapolis, MN, ASME, DE-Vol. 69-2, pp. 7-14.

[61] Chen, W., Simpson, T. W., Allen, J. K., Mistree, F., 1996, "Using Design Capability Indices to Satisfy a Ranged Set of Design Requirements", Proceedings of ASME Design Engineering Technical, Design Automation Conference, Irvine, CA, DETC1996/DAC-1090

[62] Chen, W., Simpson, T. W., Allen, J. K., Mistree, F., 1999, "Satisfying Ranged Sets of Design Requirements using Design Capacity Indices as Metrics", *Engineering Optimization*, Vol. 31, pp. 615-639

[63] Simpson, T. W., Allen, J. K., Mistree, F., and Chen, W., 1997, "Designing Ranged Set of Top-level Design Specifications for a Family of Aircraft: An Application of Design Capability Indices", 1997 World Aviation Congress, pp. 975513, Anaheim, CA

[64] Simpson, T. W., Rosen, D., Allen, J. K., Mistree, F., 1998, "Metrics for Assessing Design Freedom and Information Certainty in the Early Stages of Design", *ASME Journal of Mechanical Design*, Vol. 120, Num. 4, pp. 628-635

[65] Marston, M., 2000, "Game Based Design: A Game Theory Based Approach to Engineering Design", Ph.D. Dissertation, George Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta

[66] Murata T. 1989, "Petri Nets: Properties, Analysis and Applications", Proceedings of IEEE, 77(4).

[67] Petri, C. A., 1966, "Communication with Automata", New York: Griffiss Air Force Base, Tech. Rep. RADC-TR-65-377, Vol. 1, Suppl. 1

[68] Petri, C. A., 1963, "Fundamentals of a Theory of Asynchronous Information Flow", in Proc. IFIP Congress 62, pp. 386-390

[69] Holt, A. W., Saint, H., Shapiro, R., Warshall, S., 1968, "Final Report of the Information Systems Theory Project", Tech. Rep. RADC-TR-68-305, New York: Griffiss Air Force Base

[70] Holt, A. W., Commoner, F., 1970, "Events and Conditions", Princeton, N. J., applied Data Research Inc., Information System Theory Project, Record Project MAC Conf. Concurrent Systems Parallel Computation, pp. 3-52

[71] Holt, A. W., 1971, "Introduction to Occurrence Systems, in Associative Information Techniques", L. Jacks, Ed., New York: American Elsevier, pp. 175-203

[72] Shariro, R. M., Saint H., 1970, "A New Approach to Optimization of Sequencing Decisions", *Ann. Rev. Automatic Programming*, Vol. 6, No. 5, pp. 257-288

[73] Commoner, F., Holt, A. W., Even, S., Pnueli A., 1971, "Marked Directed Graphs", *J. Comput. Syst. Sci*., Vol. 5, pp. 511-523

[74] Commoner, F., 1972, "Deadlocks in Petri nets", Wakefield, Applied Data Research, Inc., Report #CA-7206-2311

[75] Dennis, J. B., Ed., 1970, "Record Project MAC Conf. Concurrent Systems and Parallel Computation", pp. 199

[76] Peterson, J. L., 1981, "Petri Net Theory and the Modeling of Systems", Englewood Cliffs, NJ: Prentice-Hall, Inc.

[77] Reisig, W., 1985, "Petri Nets, EATCS Monographs on Theoretical Computer Science", Vol. 4, New York: Springer-Verlag

[78] Agerwala, T., 1979, "Putting Petri nets to Work", Computer, Vol. 12, No. 12, pp. 85-94

[79] Johnsonbaugh, R., Murata, T., 1982, "Petri nets and Marked Graphs-Mathematical Models of Concurrent Computation", *the American Math. Monthly*, Vol. 89, No. 8, pp. 552-566

[80] Peterson J. L., 1977, "Petri nets", *ACM Computing Surveys*, vol. 9, No. 3, pp. 223-252

[81] Zhu, H., Zhou, M. 2002, "Formalizing the Design of a Collaborative System by Petri Nets", Proceedings of the second IEEE International Conference on Systems, Man and Cybernetics (SMC'02), October 6-9, 2002, Hammamet, Tunisia.

[82] Zhao, L., Jin, Y. 2000, "Modeling Collaborative Engineering Design Process using Petri-net", Proceedings of ASME Design Engineering Technical Conferences, September 10-13, 2000, Baltimore, Maryland.

[83] Huang, J. Y., Deng L. Y. 2000, “The Petri Net Model for the Collaborative Virtual Environment on the Web”, *Tamkang Journal of Science and Engineering*, Vol. 3, No. 4, pp. 267-281.

[84] Zhu, H. B., Zhou, M. C., 2002, “Formalizing the design of a Collaborative System by Petri nets”, Systems, Man and Cybernetics”, 2002 IEEE International Conference on Vol. 5, pp. 6

[85] Zhao, L., and Jin Y. (2000) Modeling, “Collaborative Engineering Design Process using Petri-net”, DETC2000/CIE-14609

[86] Paul, G., Beitz, W., 1996, “Engineering Design—A Systematic Approach”, Second Edition, Springer, London

[87] Suh, N. P., 1990, The Principle of Design, Oxford University Press, Oxford

[88] Hauser, J. R., Clausing, D., 1988, “The House of Quality”, Harvard Business Review, May-June, pp.63-736

[89] Yoshikawa, H., “General Design Theory and a CAD System, Man-Machine Communication in CAD/CAM”, T. Sata, E. Warman. IFIP, 1981

[90] Adler, P., and elbaum, A., 1995, “From Project to Process Management: An Empirically-Based Framework for Analyzing Product Development Time”, *Management Science*, Vol. 41, No. 3, pp.458-484.9

[91] Eppinger, D. S., 1997, “Generalized Models of Design Iteration Using Signal Flow Graphs”, *Research in Engineering Design*, Vol. 9, pp.112-123.10

[92] Bras, B. A., Mistree, F., 1991, “Designing Design Process in Decision-Based Concurrent Engineering”, *SAE Transactions Journal of Materials &Manufacturing*, Vol. 100, pp. 451-45811

[93] Bar-Yam, Y., 1997 Dynamics of complex system

[94] Cowan, G. A., Pines, D., and Meltzer, D., 1994 "Complexity: Metaphors, Models and Reality", Addison-Wesley, Reading, MA

[95] Yassine, A., and Braha, D., 2003, "Complex Concurrent Engineering and Design Structure Matrix Method", *Concurrent Engineering*, Vol. 11, No. 3, pp.165-176

[96] Kusiak, A, and Wang, J.R., 1993, "Decomposition of the Design Process", *Journal of Mechanical Design*, 115:687-696

[97] Krishnamurthy, K., Law, H. K., 1997, "A Data Management Model for Collaborative Design in a CAD Environment", Engineering with Computers, Vol. 13, pp. 65-86

[98] Chen S. J., and Lin L., 2002, "A Project Task Coordination Model for Team Organization in Concurrent Engineering," *CERA Journal*, 10(3), September, PP.187-202

[99] Sriram, D., Ahmed, S., Logcher, R., 1992, "A Transaction Management Framework for Collaborative Engineering", *Engineering with Computers*, Vol. 8, pp. 213-232.

[100] Majumder, D., Rangan, M. R., Fulton, E. R., 1994, "Information Management for Integrated Design Environments", *Engineering with Computers*, Vol.11, pp. 227-245

[101] Pahng, F., Senin, N., and Wallace, D. 1998. "Distributed Object-based Modeling and Evaluation of Design Problems", *Computer-Aided Design*, 30(6), pp. 411–423

[102] Lu, S. C-Y., and Cai, J. 1999, "Modeling Collaborative Design Process with a Socio-Technical Framework", Proceedings of Sixth. ISPE Int'l Conference Concurrent Engineering, Bath, UK

[103] Ullman, D. G., 1988, "A Model of the Mechanical Design Process Based on Empirical Data", *AI EDAM*, 2(1), pp. 33-52

[104] Sun, Q., and Gramoll, K., 2002, "Internet-based Distributed Collaborative Engineering Analysis", *CERA Journal*, 10(4), Dec., pp.341-348

[105] Huang, G. Q., and Mak, K. L., 2002, "Concurrent Engineering, Agent-based Collaboration Between Distributed Web Applications: Case Study on 'Collaborative Design for X' Using CyberCO", *Concurrent Engineering*, 10(4), pp.279-290

[106] Ramani, K., Agrawal, A., Babu, M., and Hoffmann, C., 2003, "CADDAC: multi-client collaborative shape design system with server-based geometry kernel", *Journal of Computing and Information Science in Engineering*, vol(3), pp.170-173

[107] Cheryl, V., Shapirstein, Y., Cera, C. D., and Regli W. C., 2001, "Multi-user modeling of NURBS-based objects", DETC2001, September 9-12, Pittsburgh, Pennsylvania, CIE-21256

[108] Xiao, A., 2003, "Collaborative Multidisciplinary Decision Making in a Distributed Environment", Ph. D. Dissertation, School of Mechanical Engineering, Georgia Institute of Technology, Atlanta

[109] Parkinson, A., Sorenson, C., Pourhassan, N, 1993, "A General Approach for Robust Optimal Design", *ASME Journal of Mechanical Design*, Vol. 115, No. 1, pp. 74-80.

[110] Phadke, M.S., 1989, Quality Engineering using Robust Design, Prentice Hall, Englewood Cliffs, New Jersey

[111] Yu, J., and Ishii, K., 1994, "Robust Design by Matching the Design with Manufacturing Variation Patterns", ASME Design Automation Conference, ed. B. Gilmore, September 1994, Minneapolis, MN, ASME, DE-Vol. 69-2, pp. 7-14.

[112] Chang, T. S., Ward, A. C., and Lee, J., 1994, "Distributed Design with Conceptual Robustness: A Procedure Based on Taguchi's Parameter Design", ASME Concurrent Product Design Conference (Gadh, R., ed.), Chicago, IL, ASME, Vol. 74, pp. 19-29

[113] Chen, W., Allen, J. K., Tsui, K-L, and Mistree, F., 1996, "A Procedure for Robust Design", Transaction of the ASME Journal, *Journal of Mechanical Design*, Vol. 118, No. 4, pp. 478-485

[114] Montgomery, D., 1991, Design and Analysis of Experiments, John Wiley&Sons, New York

[115] David, R., Alla, H. 1992, Petri Nets and grafcet, Prentice Hall, UK.

[116] Giesing, J. P., Barthelemy, J.-F. M. 1998, "A Summary of Industry MDO Applications and Needs", 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Sept. 2-4, 1998, NASA Langley Research Center, St. Louis MO.

[117] Chen, Z. Q., and Siddique, Z., 2003, "CORBA-based Task Requirement Driven Design System", ASME IMECE2003-42781, November 16-21, 2003, Washington, D.C.

[118] Henning, M., and Vinoski, S., 1999, Advanced CORBA programming with C++, Addison-Wesley

[119] Chen, Z. Q., and Siddique, Z., 2004, "Web-based Mechanical Engineering Design Education Environment Simulating Design Firms", ASME/IMECE2004-61318, Nov. 13-19, Anaheim, California

[120] Chen, Z. Q., and Siddique, Z., 2006, "A Model of Collaborative Design Decision Making using Timed Petri-net", ASME/DETC2006/CIE-99519, Sept. 2-6, Chicago, IL

[121] Chen, Z. Q., and Siddique, Z., 2003, "Elements and Techniques to Develop a Distributed Design Framework", ASME/DETC2003/CIE-48229, Sept. 2-6, Chicago, IL

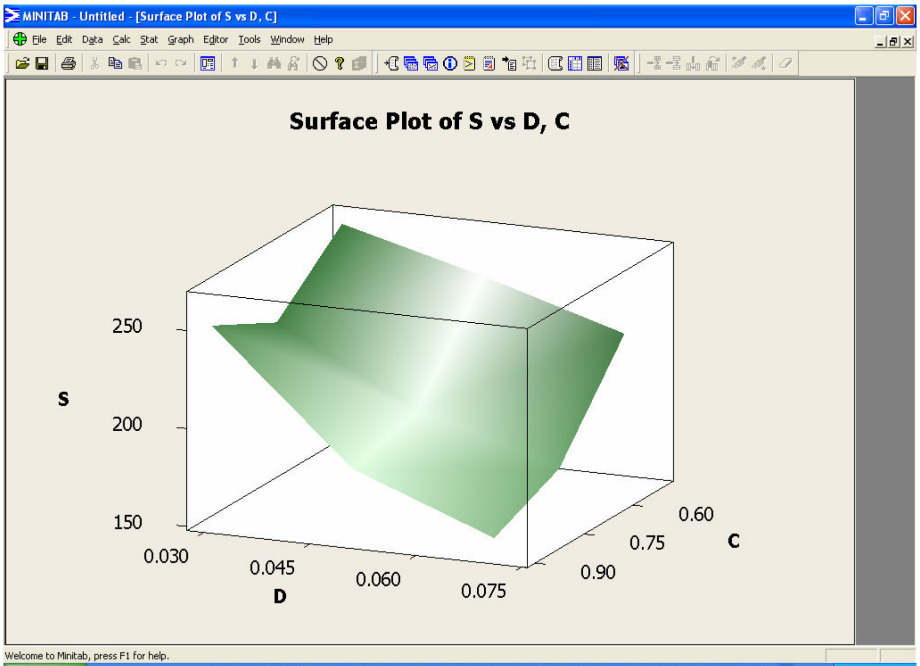
[122] Chen, Z. Q., and Siddique, Z., 2006, "A System to Support Collaboration in a Distributed Environment During Conceptual Design", *Journal of computing and information science in engineering*, 9(1), pp. 85-95

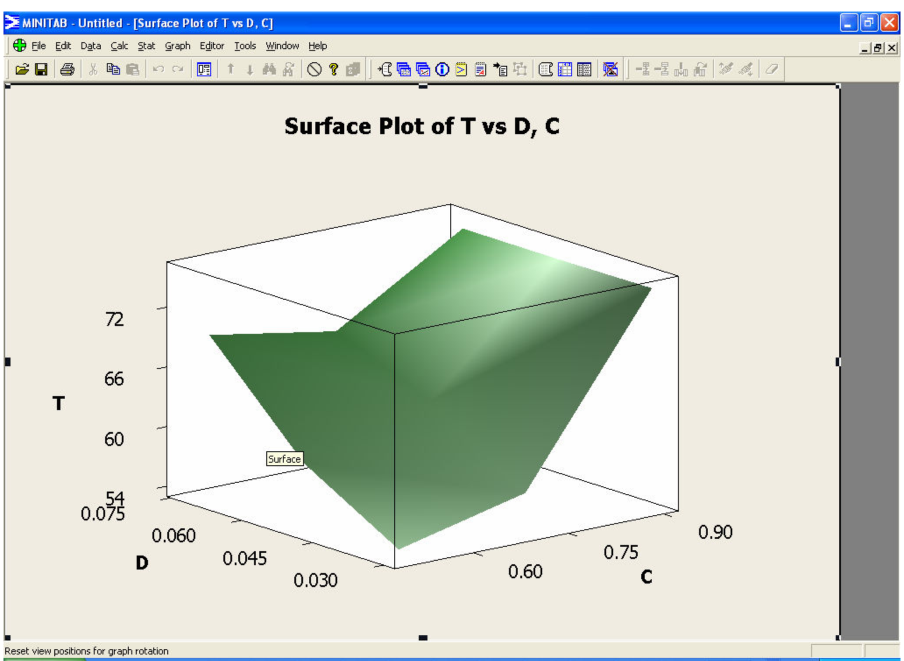
[123] Chen, Z. Q., and Siddique, Z., 2005, "A Cooperative-collaborative Design System for Concurrent Multidisciplinary Mechanical Design", CE2005: The 12th ISPE International Conference on Concurrent Engineering: Research and Applications, Ft. Worth/Dallas, USA 25 - 29 July, 2005

[124] Siddique, Z., Chang, K. H., Chen, Z. Q., and Edke, M., 2006, "A testbed to Support Collaboration among Distributed Designers for Reverse Engineering, Re-engineering and Fast Manufacturing", ASME/DETC2006/CIE-99424, September 10-13, 2006, Philadelphia, Pennsylvania

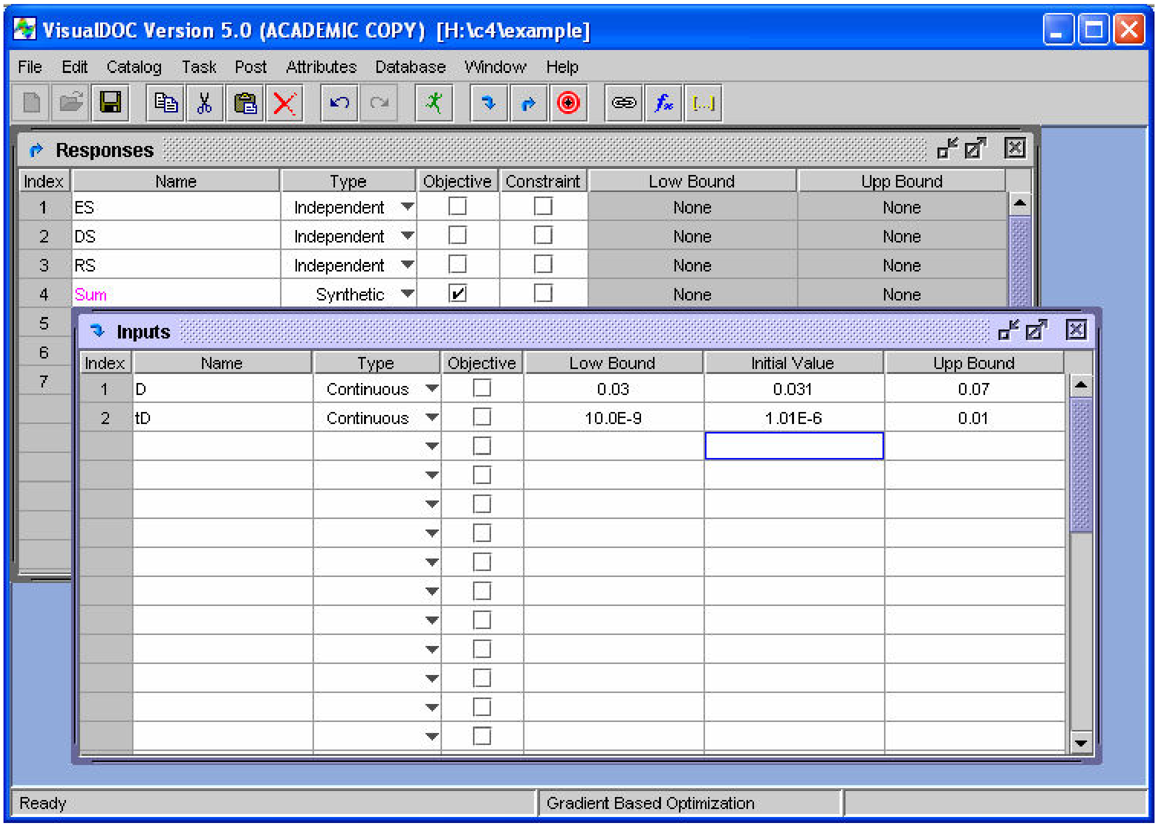
Appendix: Screen Shots and Source Code for Example in Section 4.3.3

1. Screen shot of Creating Response Surface





2. Screen Shot of VisualDoc User Interface



3. Source Code for Calculation Responses

```
#include <stdlib.h>

#include <stdio.h>

#include <math.h>

/*

//////////////////////////////////////////////////////////////////////

// Export declarations for WINDOWS

//////////////////////////////////////////////////////////////////////

*/

#if defined(_MSC_VER)
```

```
#ifdef __cplusplus
extern "C" {
#endif
void __declspec(dllexport)    UserAnalysis( int *pnNPoints, int *pnNInputs,
                int *pnNResps, double *adInputs,    double *adResps );
#ifdef __cplusplus
}
#endif
#endif
void UserAnalysis(int *pnPoints, int *pnInputs, int *pnResps,double *adInputs, double
*adResps)
  {
      /* Define some counters */
      int i, nInputsStart, nRespsStart;
      /* The design variables */
      double C = 0.0;
      double D = 0.0;
      double tC = 0.0;
      double tD = 0.0;
      /* The responses */
      double S=0.0;
      double T=0.0;
      double ConS=0.0;
```

```
double ConT=0.0;
double MaxS=0.0;
double MinS=0.0;
double MaxT=0.0;
double MinT=0.0;
double SS=0.0;
double TT=0.0;
/* The objective */
double ES=0.0;
double DS=0.0;
 double RS=0.0;
double ET=0.0;
double DT=0.0;
 double RT=0.0;
/* Loop over all points and do analysis for each */
for( i = 0; i < *pnPoints; i++ ) {
    MaxS=-1000.0;
    MinS=1000000000.0;
    MaxT=-1000.0;
    MinT=1000000000.0;
    /* Map design variable values to local variables */
    nInputsStart = *pnInputs * i;
    D = adInputs[nInputsStart+0];
```

```
tD = adInputs[nInputsStart+1];

C=0.39455+3.555*D;


S=355-113.7*C-1599.7*D+502.7*pow((C-0.7),2)-3276.8*(C-0.7)*(D-0.05)+21187.9*pow((D-0.05),2);

// calculate constraints

ConS=S+fabs(-113.7+502.7*2*(C-0.7)-3276.8*(D-0.05))*tC*1.5+fabs(-1599.7-3276.8*(C-0.7)+21187.9*2*(D-0.05))*tD*1.5;

/* Calculate ACTUAL response values */

for (double DD=D-tD;DD<D+tD;DD+=tD/10000)

    {

        double CC=C;

        if (DD>0) {


SS=355-113.7*CC-1599.7*DD+502.7*pow((CC-0.7),2)-3276.8*(CC-0.7)*(DD-0.05)+21187.9*pow((DD-0.05),2);

        if (SS>MaxS) MaxS=SS;

        if (SS<MinS) MinS=SS;

        }

    }

/* Map the local response variables to the output array */

ES=fabs(S/190.0-1.0);

DS=fabs((MaxS-MinS)/2.0/(10.0)-1.0);
```

```
    RS=fabs((MaxS-MinS)/2.0/(fabs(tD))/(1)-1.0);
    nRespsStart = *pnResps * i;
    adResps[nRespsStart+0] = ES;
    adResps[nRespsStart+1] = DS;
    adResps[nRespsStart+2] = RS;
    adResps[nRespsStart+3] = 0;
    adResps[nRespsStart+4] = MaxS;
    adResps[nRespsStart+5] = MinS;
    adResps[nRespsStart+6] = SS;
};
return;
```