Unified Interdisciplinary Methodology for Concurrent and Collaborative Engineering in Chemical Process Industry

by

Yanan Xie

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical Engineering
University of Alberta

Abstract

Chemical process engineering projects are complex and multi-disciplinary, requiring collaboration of different domains, such as business project management, chemical process, mechanical, electrical, and instrument with specific computer-aided tools. Current research on semantic interoperability is still domain specific. The lack of interoperability across domains remains a big issue in industry practice. The industrial demand for systematically managing the information generated throughout the lifecycle of engineering processes and associations across disciplines imposes a great research challenge.

Rather than exchanging documents from one department to another, such as the chemical process department to the mechanical one, this research proposes an interdisciplinary engineering methodology based on a unified informatics approach to develop a systematic technology for supporting chemical process project lifecycle. Semantic characteristics of information entities and flows across chemical and mechanical engineering domains, and the implicit associations, are discussed. The similarity between the two disciplines inspires a unified engineering framework. Under this framework, two categories of new features that can be identified from commonly-observed chemical engineering processes, associative chemical process features and inter-domain functional features, are modeled. Related to those traditional product-related features in the mechanical engineering domain, the above two sets of features offer new mechanisms to support a multi-disciplinary and feature-based chemical process modeling system. Based on the proposed feature models, interdisciplinary engineering information association mechanisms are constructed. Such interdisciplinary engineering associations are explicitly expressed and systematically managed by constraint models. Hence, the feature associations are well-maintained along the project life cycle. A mechanism is then developed to dynamically construct feature parameter association map, which provides context association information among engineering entities. Based on the generated map, a well-controlled, incremental and dynamic engineering change propagation method is proposed to assist engineers with an intelligent change propagation solution.

This proposed unified engineering methodology offers a solution of comprehensive and feature-based system modeling for real-world complex problems of system integration and interoperability, and hence, is capable of supporting engineering collaboration across disciplines. The insights gained by this research work also add to the growing understanding of relationships among engineering design in separate disciplines. Implementation of a prototype system based on feature definition and consistency maintenance mechanisms leads to a collaborative engineering platform for the chemical process design, which provides a feature-based modeling to explicitly represent characteristics of engineering significance as well as such associations. Thus information sharing is facilitated, while the feature models and constraints are all systematically managed. The prototype of applications of the proposed features shows the effectiveness towards consistency and efficiency improvement for chemical engineering informatics modeling. The mechanism proposed is capable of maintaining a consistent design through the life cycle of the chemical process project, and hence, the efficiency can be potentially improved by reducing the tedious revision work led by inconsistent design.

Preface

This thesis work is completed by Yanan Xie under the supervision of Dr. Yongsheng Ma. The overall research direction was suggested by Dr. Yongsheng Ma; while the detailed research topics, modeling, proposed methodology, prototype development and paper writing were my original work. The contents of each chapter have been/will be published in the journals or conference proceedings.

List of Publications:

1. Xie, Y., & Ma, Y. (2015). "Design of a multi-disciplinary and feature-based collaborative environment for chemical process projects", *Expert Systems with Applications*, Elsevier, 42(8), 4149-4166.
2. Xie, Y., & Ma, Y. (2014). "Unified Interdisciplinary Methodology for Collaboration in Chemical Process Industry", *Proceedings of the 8th International Conference on Foundations of Computer-aided Process Design*, reprinted in Computer-Aided Chemical Engineering, 34, Elsevier, 747-752.
3. Xie, Y., Wei, J. & Ma, Y. (2012). "Multi-view Feature Model Representation to Support Integration of Chemical Process and Mechanical Design", *Computer-Aided Design and Applications*, 10(4), Taylor & Francis, 619-628.
4. Xie, Y., & Ma, Y. "Well-Controlled Engineering Change Propagation via a Dynamic Inter-Feature Association Map", submitted to the journal, *Research in Engineering Design*.
5. Xie, Y., & Ma, Y. "Unified Interdisciplinary Constraints Association Modeling", to be submitted.

6. Xie, Y., Liu, J., Liu H., & Ma, Y. (2013). "Features and interoperability of computer aided engineering systems", In *Semantic Modeling and Interoperability in Product and Process Engineering*. London: Springer.
7. Sajadfar, N., Xie, Y., Liu, J., Liu H., & Ma, Y. (2013). "Introduction to Engineering Informatics", In *Semantic Modeling and Interoperability in Product and Process Engineering*. London: Springer.
8. Sajadfar, N., Xie, Y., Liu, J., Liu H., & Ma, Y. (2013). "A Review of Data Representation of Product and Process Models", In *Semantic Modeling and Interoperability in Product and Process Engineering*. London: Springer.

Chapter 2 is based on combination of literature reviews from papers 1, 2, 4 and 5.

Chapter 4 is based on paper 1.

Chapter 5 is based on papers 1, 2 and 3.

Chapter 6 is partly based on 4. The majority part of this chapter contributes to paper 5, which is to be submitted to a peer reviewed journal.

Chapter 7 is based on paper 4.

Chapter 8 is based on a combination of case studies in each of the published articles.

Acknowledgement

Here, I would like to take this opportunity to express my appreciations to a number of remarkable people. Without them, this work would not be presented here.

First, I would like to thank my supervisor, Dr. Yongsheng Ma, who is my mentor to academia and life. He has provided me with lots of valuable advice in the past five years, which will benefit me all my life. Enthusiastic discussions with him always inspire excellent ideas. Also, he always encourages me to do the best in my work. Thanks to him, I am able to develop myself and produce this work.

I am grateful to CSC, McCoy, NSERC, and Mitacs for their generous financial support for my research. Also, NSERC and Mitacs provided me with three years of industrial experience as an intern in McCoy, getting me familiar with the industrial environment in Canada. Here, I would like to thank my colleagues in McCoy: Abdul, Jose, and Luis, who gave me lots of guidance and direction on how to work in such an environment. My thanks also go to all my colleagues in our research group, who contribute to a friendly research environment. Among them, I would particularly like to thank my friends, Zhengrong Cheng, Jikai Liu, Wenjing Wang, Lei Li and Junyu Fu, who has also helped me a lot in everyday life.

I would like to acknowledge my committee members, Dr. Michael Lipsett, Dr. John Doucette, and Dr. Arno de Klerk, who provided valuable suggestions for my research and changed their schedules to make the oral defense happen. Special thanks to Dr. Deyi Xue, who took a long trip from University of Calgary to attend my oral defense.

Last but not the least, I appreciate the support from my parents and my wife. It’s you who are always around me. Whenever I feel rotten, you encourage me to believe in myself and get out of depression. Your constant love and support encourages me to go through all the challenges. Also, my two little ones bring lots of joys and colors to my everyday life and make me believe my future life is full of hope. I would like to devote my thesis to you. Thank you all.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Background

A chemical process engineering project can be characterized as a series of complicated, multi-phase, interdisciplinary design and engineering activities that involve interdependent contributions from chemical, mechanical, and electrical engineers. The complexity of such projects has led to the development of computer-aided software packages. Such computer-aided tools provide engineers with powerful and intelligent functions and capabilities for individual engineering areas, such as chemistry analysis, chemical process design, and mechanical design. For example, Aspen Technology packages are commonly used for chemical process conceptual design and optimization; Intergraph SmartPlant^{TM} for chemical process and plant design; and Siemens NX^{TM}, Dassault Catia^{TM}, and Solidworks^{TM} for mechanical design.

The keen market competition calls for a short lead time for any chemical process projects, with zero tolerance for faulty design. Engineers from different disciplines need a collaborative design and engineering environment to work effectively and efficiently. This strongly demands systematic modeling of associations among the entities generated from different software packages due to the dependencies among engineering activities. For example, the conceptual chemical process analysis provides input to the detailed chemical process engineering, and it also influences detailed mechanical design, as shown in figure 1.1. Unfortunately, those aforementioned software tools are developed by different vendors emphasizing individual domains of engineering, and hence have distinct semantics and data structures. While such individual tools are widely applied, there is still no integrated solution for industry use; a systematical method to manage the proprietary data generated by such software packages is missing. One of the reasons is that there are no development incentives for those software vendors, who try to lock in customers with their proprietary design models (Hoffmann, 2005). Also, the associations among engineering entities are not fully identified yet, and hence no generic solutions could be easily given.

Current industry suffers from such interoperability problems led by various software packages used by different domain engineers, and the resulted loss is unbelievably huge. It was reported by NIST (Gallaher et al., 2004) that an annual cost of more than $15.8 billion has resulted from inadequate interoperability among the computer aided software systems in the capital facilities industry. The lack of systematic management of multi-disciplinary engineering data even makes it inconvenient and time-consuming for domain engineers to find context information. This situation leads to collaboration difficulties among engineering disciplines, as well as the corresponding tools, due to the difference of specific syntax and semantics embedded in the proprietary data representations (Schneider & Marquardt, 2002).

Owing to the lack of a common infrastructure support for collaborative engineering, three urgent problems hinder the collaborations of domain engineers. First, there are no well-defined mechanisms to pass the information from one domain to another—for instance, from the chemical process design domain to mechanical design—and hence, information transfer is tedious and error-prone while most of engineering semantics are skipped during the transfer of documents. The same problem exists when feedback information is passed from a downstream engineering domain to an upstream one. Second, costly problems commonly occur due to the lack of the anticipation for downstream engineering considerations in the early conceptual design stages. Last but not least, there is no management of association among engineering entities, and hence change impacts are difficult, if not impossible, to identify.

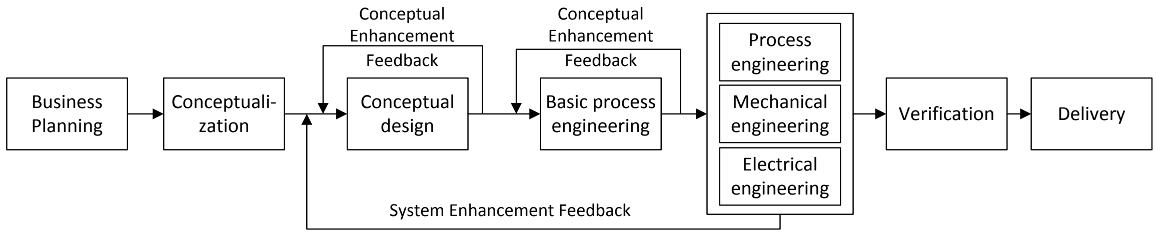


Figure 1.1 Typical activities involved in the lifecycle of a chemical process project

For example, without integrated collaboration information support, the conceptual process design could be neither viable at the early design stage because of equipment constraints nor accurate without cycles of revisions that are made necessary due to the current pieced-information feeding practice. It is also very costly in time and effort to verify and change inter-disciplinary design models according to the current practice. For example, the designed process operating parameters, such as temperatures and pressures, are the original drivers for the downstream mechanical design and engineering phase. Yet, these operating parameters are dependent upon or closely associated with the mechanical design and selection of best-fit equipment items or components with economic and safety considerations. Currently, the changes of those driving parameters lead to significant redesign efforts in the project. During the mechanical design process, the equipment items are usually classified into two categories: proprietary equipment (PE) items and non-proprietary equipment (NPE) items (Towler & Sinnott, 2013). For those NPE items, such as pumps and compressors, it is more economical for the owners or the engineering procurement and construction (EPC) companies to get them from those off-the-shelf products instead of making the specially-designed ones. The off-the-shelf equipment items can be supplied and produced in batches. For example, an off-the-shelf vessel that is larger than the volume requirement usually costs less than a customized vessel with the exact size of the volume requirement (Couper et al., 2010). In contrast to the NPE items, the PE items warrant more collaboration between chemical process and mechanical engineers. To keep the data consistent, based on the input of chemical process engineering, mechanical engineers design the PE items cyclically. After determining the PE item design, the PE item's new mechanical properties or specifications have to be used to update the parameters in the chemical process model. The new process model needs to be further verified by mechanical engineers.

In the current industry practice, such cyclical work is done manually by engineers in EPC companies. One disadvantage of traditional design is that the deliverables are isolated, separated documents, including drawing, specification sheet, equipment list, etc. Whenever a design change is needed, e.g. in a plant revamp, no data support is available. The impact of design change cannot be predicted and controlled, which can easily lead to high risk. Also, design results in the document format cannot be transferred to valuable knowledge assets, as they are expected to, since these results are tedious to be referenced or reused when designing a similar project. It is cumbersome, time-consuming, and error-prone. Many errors are hard to detect, which, more often than not, leads to problems in the testing and even operating phases. To design a multi-disciplinary engineering collaboration environment, the associations among different engineering entities of different disciplinary domains need to be explicitly represented and systematically managed; also, it calls for a systematic propagation method for numerous expected design changes.

So far there have been few research efforts reported on developing a multi-disciplinary design engineering environment to associate different engineering domains, despite the tremendous potential for quality improvement and cost reduction. Most researchers still mainly consider domain-specific problems; they have focused on either chemical process design or mechanical design of the process equipment items. The system interoperability problem remains without a unified framework, as well as explicit representation and management of the intricate associations.

1.2 Research Goal and Scope

From engineering management point of view, engineering jobs are disciplinarily separated and assigned into departments with respect to the specialties in certain areas. However, in a complex chemical engineering project, such boundaries are getting blurred, as is discussed in section 1.1. The engineering results generated from each discipline need to comply with others to work coherently. Rather than throwing the paper/electronic documents from one department to another, this work aims to propose an interdisciplinary engineering methodology based on a unified informatics approach to develop a systematic technology for supporting semantic information sharing across project lifecycle. Here, the proposed feature-based model is applied across the whole engineering domain rather than individual discipline models. The characteristics of the engineering entities are abstracted into hierarchical semantic feature model, and the associations among those entities are also explicitly defined and maintained from the start to the end of the project. With a supporting change propagation method, the data are expected to be kept consistent all through the life cycle of the chemical process projects.

- A generic and flexible data representation schema across the whole engineering domain

A semantic comparison across engineering disciplines needs to be performed to identify the commonality that they share, and hence the engineering semantics can be abstracted into a schematic model. Such a schematic model should be generic enough to represent the multi-facet characteristics of the engineering entities across different engineering disciplines. A preliminary step is to identify the common feature objects or "engineering patterns," and list those static attributes in the proposed schematic model. Especially, the associations between the feature model and those information models are to be determined. Such information will provide a quantitative reference to future design after it is stored in the design history library. After that, dynamic behaviours and phenomena events can then be logically distilled into functions (methods) and embedded into sustainable semantic models to support the lifecycle of engineering projects. The associative feature technology is adopted as the foundation and a set of comprehensive features are defined. A set of associative chemical process features are proposed to abstractly represent the chemical process in a hierarchical way; while inter-domain functional features provide explicit association representation between chemical process and mechanical design features as well as functional explanation and corresponding contexts. Along with mechanical features, they provide a unified feature-based modelling scheme for chemical engineering projects. However, the feature definition as well as its application in mechanical domain will not be the focus of the research. Rather, those well-established definitions will be adopted as the components of the unified engineering informatics.

- Interdisciplinary constraint association modelling

Complex associations commonly exist in the engineering design of chemical plant projects, especially among the features across disciplines. For example, the operating pressure of a reaction process should be associated with the design pressure of the reactor. Such association is reflected as a constraint that the design pressure of the mechanical design feature should be higher than the operating pressure of the chemical process feature. This kind of constraint associations collectively represent engineering intent; while further processing and reasoning requires explicit representation of such associations (Ma & Bong, 2010). Constraints, which explicitly express the engineering associations, have to be modelled and used to validate the consistency of features (Ma & Tong, 2003), and consequently data consistency can be maintained along the lifecycle of chemical process projects. Engineering constraints include design rules or codes, engineer preferences/assumptions, constitutive equations, inter-feature constraints across stages, etc. The systematically management of constraint association modelling is expected to provide a data basis for the design change propagation mechanism proposed.

- A scalable and dynamic design change propagation method

Design changes are inevitable in engineering processes. One of the main reasons leading to the delay of project schedules is that there is no efficient way to handle the design changes, especially in a complex interdisciplinary environment. Due to the intricate associations in a chemical process project, such changes lead the design to be error-prone and inconsistent, as each design change could easily lead to a chain effect on other entities. For example, changes to operating conditions of a unit process design in a specific part usually induce adjustment in the process equipment, even significant structural changes (Nagl & Marquardt, 2008). Such snowball change propagation requirement calls for an effective system that can provide sophisticated change propagation based on intelligent analysis of the constraint network so that the design changes suggested have the least impact on the existing design while resolving engineering conflicts. Consequently, a well-controlled change propagation mechanism, namely *progressively expanded constraint satisfaction problem* (PECSP) solving method, is presented under the unified feature-based interdisciplinary engineering framework.

1.3 Road Map of the Thesis

This chapter introduces the general context of chemical process projects as well as the problems that exist in current industrial practice owing to heterogeneous models in individual disciplines. Especially, complex interdisciplinary associations were not well maintained. Also, impacts from common engineering changes are difficult to identify and handle; this can easily lead to tedious and error-prone rework and hence delay of project delivery. The motivation of this thesis is to solve such informatics interoperability issues.

The remainder of the thesis is organized as follows.

Chapter 2 summarizes literature review on integrated product and process modeling in chemical and mechanical engineering domains, and introduces advances in engineering change propagation and constraint satisfaction problems, as well as feature technology.

Chapter 3 develops a feature-based modeling schema related to chemical process projects. The feature technology is introduced into chemical domain from mechanical domain. The developed chemical process feature schema offers a comprehensive, hierarchical and flexible representation of engineering semantics of chemical processes in different granularities. Along with well-established mechanical features, a generic data representation across all engineering domains of chemical process projects is provided.

Chapter 4 proposes a unified interdisciplinary engineering framework based on the explorations of commonalities and associative relationships of product and process models between chemical and mechanical domains. The invocation of such unified framework is that inter-domain association and dependencies are explicitly represented and well maintained by the proposed inter-domain functional features, which offer a flexible establishment and explicit representation of associations among domain-specific features.

Chapter 5 investigates interdisciplinary associations in chemical process projects and develops a constraint-based model of such associations. The explicit representation of engineering constraints offers a quantitative evaluation capability rather than just qualitatively register the dependencies among detailed engineering model entities. Also, with systematic management of feature and constraint models, the association information evolves along with the life cycle of projects. A mechanism to dynamically generate a feature parameter association map is provided. The advantage of the generated map is that it always reflects the most updated dependency information, which provides a data basis to the proposed change propagation method in chapter 6.

Chapter 6 proposes a progressive expanded constraint satisfaction problem solving method to effectively propagate engineering changes while eliminating engineering conflicts. Based on reasoning on the dynamically generated association map, the PECSP method automatically searches for connected parameter and constraint nodes, and formulates constraint satisfaction problems (CSPs) with progressively expanded scope. This will limit the size of the formulated problem and hence the computation efforts. What's more, the generated solution tends to make fewer modifications to previous engineering design to absorb the change impacts.

Chapter 7 modularizes the proposed methods in a prototype system developed to validate that the proposed method can effectively improve engineering consistency across disciplines by intelligently detecting and solving engineering conflicts.

Chapter 8 concludes the research with highlights of the contribution of the research and proposes some recommendations for future work.

2 Literature Review

2.1 Introduction

This review starts with the state-of-the-art of integrated product and process modeling in both the chemical engineering and mechanical engineering domains. The limitation of the current existing research is discussed; and a research gap is identified, which will be the main scope of this research.

Then, advances in engineering change propagation are addressed in section 2.3. So far, the evaluation of engineering change impacts cannot be done quantitatively; the information granularity represented by commonly adopted design structure matrix is too coarse. Also, such association matrix cannot evolve along with the lifecycle and hence it will be tedious to maintain. Section 2.4 introduces common techniques of constraint satisfaction problem solving and some of software packages are briefly discussed. In section 2.5, advances of traditional features as well as their applications are firstly given. Since the feature technology firstly proposed in 1980s, feature-based design has been a proven method to support concurrent and collaborative engineering in the mechanical domain. Then an advanced feature modeling mechanism, associative feature, is introduced. Finally, major components of associative feature and constraint modeling in feature based design scheme are discussed.

2.2 Integrated Product and Process Modeling

As shown in figure 1.1, the engineering design of a chemical process project is a multi-disciplinary engineering process to transform the business concept into operation. The scale of a project and those numerous close associations among the entities and processes within and across the disciplines make the requirement of a unified engineering system a great challenge, and hence, should draw great attention to the academic research.

2.2.1 Integrated Product and Process Modeling in Chemical Domain

Advances of the research to integrate chemical product and process modeling can be represented from a systemized engineering point of view (Eden et al., 2004; Stephanopoulos & Reklaitis, 2011). For example, one of the efforts was to link the molecular product and separation process design by reverse problem formulation (Eden et al. 2004). In Gernaey's work (Gernaey & Gani, 2010), a systematic model-based approach is proposed to take into consideration both pharmaceutical product and process design. In these works, the interactions between chemical product design and conceptual chemical process design have been modeled by constraint modeling. A feedback mechanism was provided to bridge these two aspects.

The computer-aided process engineering (CAPE) community has been working on the integration effort from the conceptual process design to detailed engineering. One important set of standards that was defined in CAPE Open for integration in this field is for the interoperability among various computer-aided process engineering programs (Charpentier, 2010; Jaworski & Zakrzewska, 2011). The CAPE Open standards adopt an object-oriented approach, which views each individual process modeling component as a separate object, and handles the communications among these objects by middlewares (Morales-Rodríguez et al., 2008). Based on CAPE Open standards, Zitney developed the "Advanced Process Engineering Co-Simulator," which combines steady-state simulation with equipment simulations (Zitney, 2010). Such simulation tools, along with commercial simulation software, such as Aspen hysys and PROSIM, provide quantitative insight about physico-chemical behaviors taking place in the unit equipment. There are also some other research efforts from the abstracted, conceptual modeling level. Marquardt and Nagl made a work-process-centered integration attempt in the IMPROVE project in a posterior fashion (Marquardt & Nagl, 2004). With the development of formal ontology for the CAPE domain (Morbach et al., 2009), named OntoCAPE, it has been applied to semantically integrate heterogeneous data across the process engineering phases, from conceptual to detailed engineering (Wiesner et al., 2011).

2.2.2 Integrated Product and Process Modeling in Mechanical Domain

In the mechanical engineering domain, extensive research has also been conducted on the system integration from product conceptual design to downstream phases, such as manufacturing process planning. Such integration across the product life cycle relies on the unification of product and process models of various application systems. For example, computer-aided design (CAD) and computer-aided process planning (CAPP) integration entails the interpretation and information transfer from product models to process models to facilitate the process planning activities, which involve process selection, operation sequencing as well as the allocation of machining equipment for each operation (Mokhtar & Xu, 2011; Zhou et al., 2007). To facilitate such integration, the Standard for the Exchange of Product model data (STEP) (ISO 10303) standard was proposed to provide a comprehensive set of neutral product information. A recent review of STEP can be found in Xie et al. (2013). However, STEP suffers from the rigidity and complexity in implementation and was not intended to share design intents, such as design history and constraints (Ma & Tong, 2003; Xie et al., 2013). To address the above mentioned problems, unified feature technology is introduced and has now been widely accepted as an effective tool to handle the interoperability problems (Ma et al., 2008). Historically, two main research streams are feature recognition and feature-based modeling (Babic et al., 2008; Chu et al., 2012; Hayasi & Asiabanpour, 2009). Rich information is represented with feature models, and hence the considerations of downstream issues in early design phases are improved. Some attempts have also been made to implement integrated product and process modeling in a web-based environment for remote manufacturing based on feature technology (Alvares & Ferreira, 2008). Exploration of the feature technology is further illustrated in Section 3.

2.2.3 Limitations of the Existing Research

The interoperability problem across the product and process domains is well-recognized in both chemical and mechanical engineering. However, the existing research works in the product and process integration are still mostly domain-specific. For example, some research on chemical reactor modeling and design is still carried out from the chemical process point of view, although similar terms like from mechanical design are used (Luyben, 2010; Jarullah et al. 2012). These research efforts provided fundamental input elements for further mechanical design and engineering, but strictly speaking, they were more focused on the reaction rather than the reactor design. These works investigated the reactions in certain conceptual reactor modes without touching any mechanical details of the reactor. In most chemical engineering projects, such as those for oil sands upgrading in Canada, the EPC companies deal with the design of chemical processes, as well as the mechanical products that are required to support the operation of such chemical processes. Unfortunately, there lacks a unified representation for both chemical processes and mechanical products, hence the coherent interactions among different domains are not facilitated.

The scope of this research is to propose a unified interdisciplinary engineering framework, under which associative feature technology is applied, to provide a computer-interpretable information representation for both chemical processes and mechanical products, as well as to handle the associations among engineering entities across domains.

2.3 Engineering Change Propagation

Engineering changes, sometimes named design change, have been extensively studied in the past two decades (Jarratt et al., 2011). Generic business activities needed to implement change propagation in an enterprise have been proposed by Jarratt et al. (2004). The key step to the success of implementing such procedures is to identify the possible impact of engineering changes (Ouertani, 2008).

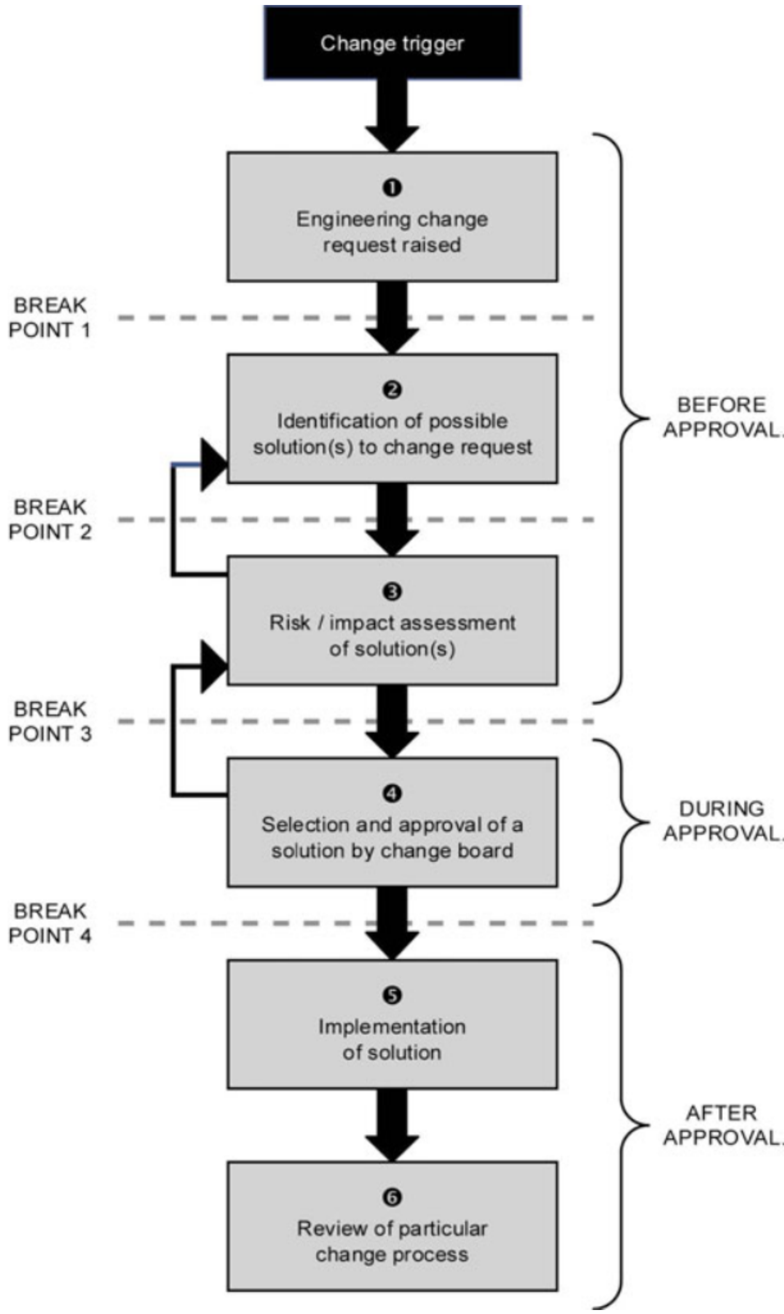


Figure 2.1 A generic model of change management process (Jarratt et al., 2004)

To identify impacts of an engineering change, association information among engineering entities is a prerequisite. Design structure matrix (DSM) as well as its variances, has been proposed to represent dependencies among different system elements (Hamraz et al., 2013; Jarratt et al., 2011). In the early days, the cross sign was commonly used in the matrix to indicate a dependency relationship between the system element/design task in the row and the one in the column (Browning, 2001). A binary value matrix is another commonly applied representation, in which 1 means that change to the element in the row would lead to the change of that in the column and 0 means the opposite. This method could facilitate algebraic calculations. Later on, extensive research efforts were spent on enriching the information represented by DSMs. For example, the intensity of the dependency relationships can also be reflected within the matrix by replacing the binary elements with numeric values. Further, Clarkson et al. (2004) refined the model with a finer granularity of information, i.e. likelihood and direct impacts of change propagation from one system element to another. The impact is defined as the average portion of the design work needed to be redone if the change propagates to this component. Such refinement would facilitate the domain experts giving an accurate estimation of the number to be put in the likelihood and impact matrices. These two matrices would then result in a direct risk matrix. A combined risk dependency structure matrix, which took into consideration the impacts resulting from both direct and indirect changes in order to predict change propagation risks, can then be derived (Keller et al., 2009). A typical DSM used nowadays is shown in figure 2.2. Generally, in a DSM, the dependencies are only recorded and analyzed in the coarse granularity, such as at the component level (Koh et al., 2012).

System	No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Bare fuselage	1		54	84	45	45	29	36	36	32	77	51	34	13	14	52	29	28	51	31
Fuselage additional items	2	4		29	10	10	6	7	8	7	23	12	7	3	3	12	6	6	12	7
Engines	3	1	1		4	4	2	4	2	1	14	4	1			3	1	1	11	1
Transmission	4	3	9	28		9	7	9	7	5	21	12	6	2	2	20	6	4	11	5
Main rotor blades	5	1	3	17	9		15	7	4	1	13	8	2		1	8	2	1	18	1
Main rotor head	6	2	6	19	9	10		6	5	3	14	9	4	1	1	18	4	3	10	3
Tail rotor	7	1	3	16	8	9	5		4	2	12	8	2	1	1	6	2	2	14	2
Hydraulics	8	6	17	40	17	15	9	13		9	32	19	10	4	4	19	9	8	17	9
Fuel	9	5	16	40	13	13	8	10	10		33	16	10	3	3	16	8	7	16	8
Engine auxiliaries	10	1	2	11	3	3	2	2	2	1		4	1			3	1	1	3	1
Flight control system	11	3	9	29	12	12	8	10	8	5	23		7	2	2	13	5	5	12	5
Avionics	12	10	25	58	23	24	14	18	17	15	48	28		6	7	39	15	13	28	15
Auxiliary electrics	13	10	29	56	24	22	15	17	17	15	45	27	17		7	27	13	14	27	15
Cabling and piping	14	34	64	93	56	56	37	46	46	40	86	62	44	20		64	41	39	63	42
Weapons and defence	15	2	5	13	4	3	2	2	3	3	10	5	3	1	1		4	2	4	3
Equipment and furnishings	16	8	23	48	18	17	10	13	15	14	39	22	14	5	5	22		12	21	13
Fire protection	17	5	16	40	13	13	7	10	10	10	32	16	9	3	4	16	8		16	8
Ice and rain protection	18	1	2	15	6	7	3	6	2	1	11	5	1			4	1	1		1
Air conditioning	19	8	22	47	18	17	10	13	14	12	38	21	13	5	5	21	11	11	21	

Figure 2.2 Example of a typical design structure matrix (Hamraz et al., 2013)

To support engineering change in a multi-company collaborative environment, Rouibah and Caskey (2003) proposed to represent dependencies among key functional parameters by a parameter relationships matrix, in which a binary number is to illustrate how close the two parameters are ("1 is for direct relationship, 2 is for 1st order indirect" and so on). This method has drawn high evaluation from industry. However, this method relies on product data management (PDM) systems, which are rather file management systems without support for relationships to fine granularity data. Hence, the information in the parameter relationships matrix is still too coarse to effectively support engineering change propagation.

Salehi and McMahon (2011) proposed a parameter structure matrix to represent geometrical feature parameters linkage information in CAD models while ignoring details of the associations. Bouikni et al. (2006) developed a product feature evolution model to coordinate engineering changes across mechanical design, manufacturing and CAE disciplines. Impacted disciplines were determined by a shared feature table containing binary relations between features and disciplines and then domain engineers were involved in the negotiation process to evaluate and to implement changes to engineering models.

So far, the above association networks require manual input of dependency information from domain experts, which is time consuming and tedious, especially for complex projects involving thousands of dependency relationships. Moreover, these association models are hard to maintain because they cannot automatically evolve along with the engineering design life cycles. Absence of management of associations between the DSM and engineering entities in current computer-aided systems prevents effectively supporting change propagation since engineering designs are typically solidified in digital models in heterogeneous software packages, such as NX and Catia. Once a design change has been implemented, there is a great chance that the given data of DSM cannot remain valid and has to be manually amended by engineers (Jarratt et al., 2011). Besides, the research efforts reported so far are limited to rather qualitative analysis of change impacts covering only process management of change propagation due to the coarse grain information representation for the associations. No substantial intelligent advice could be provided without the quantitative analysis of change effects.

This thesis aims to model the associations among engineering entities in a finer granularity based on a feature-based modeling approach and proposes a systematic method to manage such associations. Ideally, based on a dynamic association model, change impacts can be evaluated quantitatively, and consequently, a change propagation method can be developed.

2.4 Constraint Satisfaction Problem

Many engineering design problems can be modeled as constraint satisfaction problems, composing a set of variables, constraints and domains specifying possible values for each variable (Brailsford et al., 1999; Cruz & Barahona, 2005). The types of variables can be either discrete or continuous; while the domain size of the variables could be finite or infinite. It is to be noted that even discrete variables can have infinite domains. Different strategies were employed to deal with different cases to get a better efficiency.

2.4.1 Discrete Constraint Satisfaction Problem

Generally, discrete CSP solving is an NP complete problem. Backtracking search is one fundamental depth first search algorithm, which guarantees finding a solution if one exists. It generates a search tree of the solution space and abandons the branch and backtracks to the parent node as soon as the candidate solution is not a valid solution against all the constraints. Although search space is reduced when backtrack happens, still, this method struggles with large-scale problems due to the explosive solution space (Pang, 1997). The complexity of this search algorithm greatly depends on the domain sizes of variables and the number of variables. Considering that the search algorithm assigns n variables and the branching factor is d at most, thus the complexity is exponential in the worst-case scenario and the upper bound of its time complexity can be $O^*(d^n)$ (Razgon, 2005).

Since the complexity of the CSP solving depends on the domain sizes and number of variables, one way to improve solving performance is to reduce the domain sizes by enforcing arc consistency or path consistency. Maintaining arc consistency was one of the constraint propagation mechanisms proposed to filter the domain of future variables by removing inconsistent values from the domains of those variables rather than checking consistency with past variables (Van Beek, 2006). There are some other algorithms proposed to improve the efficiency of CSP solving. For example, backjumping algorithm jumps back directly to the cause of the conflict, and hence the search space can be significantly reduced (Pang, 1997).

Several heuristics have also been extensively studied to improve the efficacy of the algorithm. The minimum remaining values heuristic is one of most prevalent added in the search strategy and has been implemented by several commercial solvers, such as Gecode (Gecode, 2015). It always chooses the variables with fewest allowable values rather than selecting the next unassigned variables in the variables stack. This has been proved to be capable of significantly reducing the number of consistency checks to solve the problem (Russel & Norwig, 2010).

Local search is another fundamental paradigm for solving constraint satisfaction problems (Hoos and Tsang, 2006). The basic idea underlying local search is to start with an initial search position, which is either a randomly or heuristically generated candidate solution to the problem, and then iteratively improve the variable assignment at each step, typically based on certain heuristics.

One well known local search algorithm to solve CSP is min-conflict heuristic, which iteratively improves the assignment to a single variable in the direction of minimizing the number of unsatisfied constraints. Hence, min-conflict algorithm relies on a good initial assignment of variables, in which case a solution can be quickly approached. Also, min conflict can easily get stuck to local minima. This limitation can be overcome by introducing some mechanism to escape from local minima, e.g., tabu search or random walk. By introducing a random walk mechanism, there will be a possibility, say p ($0<p<1$), to perform a random walk rather than selecting an assignment to minimize the conflicts (Hoos & Tsang, 2006).

When combining with tabu search, the previous assignment of that variable will be declared tabu for a certain steps. In this method, the tabu tenure parameter, which defines how many steps one assignment is declared tabu, is the key parameter. Setting it to 2 was proved to consistently lead to a good performance (Stützle, 1998).

2.4.2 Continuous CSPs

CSPs over continuous variables are common in the engineering design field. One major challenge of continuous CSP solving is that continuous domains are not computer representable; hence, the solution space was approximated as boxes of Cartesian products of intervals (Pelleau et al., 2014). The boxes are generated with a branch-and-prune algorithm, which involves at least two iterative procedures (Granvilliers & Benhamou, 2006; Vu et al., 2009). The first one is branching, which splits a box into a set of smaller boxes whose union is equivalent to the initial one; the other one is pruning, which narrows down an interval box by eliminating inconsistent sub-boxes.

Targeting to continuous CSPs, some variant consistencies were proposed. Similar to arc consistency and path consistency, hull consistency and box consistency are proposed to handle the elimination of solution space in continuous CSP field (Cruz, 2005; Trombettoni et al., 2010). Constraint propagation based on these interval consistency techniques are commonly implemented in the pruning procedure to reduce the interval domains, which hence accelerates the solving process. Such interval consistency techniques rely on the interval arithmetic; some basic mathematical operations of interval arithmetic are shown as follows.

For two intervals of [a, b] and [c, d],

$$[a,b] + [c,d] = [a+b, c+d]$$

$$[a,b] - [c,d] = [a-b, c-d]$$

$$[a,b] \times [c,d] = [\min(a*c; a*d; b*c; b*d), \max(a*c; a*d; b*c; b*d)]$$

$$\frac{[a,b]}{[c,d]} = \left[\min\left(\frac{a}{c}; \frac{a}{d}; \frac{b}{c}; \frac{b}{d}\right), \max\left(\frac{a}{c}; \frac{a}{d}; \frac{b}{c}; \frac{b}{d}\right)\right]; \; when\ 0 \notin [c,d]$$

Advances in interval and continuous constraints research pushed application in engineering problems. Qureshi et al. (2010) developed an algorithm based on interval arithmetic that is proposed for solving the CSP formulated from product design, which takes into account the uncertainty associated with product model, to find a robust design solution. However, transformation from quantified notion to interval analysis has not yet been automated. Cruz and Brahona (2005) further incorporate differential equation constraints.

Recently, research interests tend to the CSPs involving mixed (discrete and continuous) variables (Gelle & Faltings, 2003; Granvilliers & Benhamou, 2006; Schichl et al., 2013). Gelle and Faltings (2003) integrated the solving process in a single search by proposing mixed refine operators to perform constraint propagation between continuous variables and discrete ones. The solver developed by Granvilliers and Benhamou (2006) treated integer variables as continuous variables but truncations of bounds were performed for those integer variables inside the interval domains in each domain reduction. The latter strategy is more commonly adopted by current available software packages.

2.4.3 Software Packages

Constraint solving is out of the scope of this research. In this thesis, a PECSP change propagation mechanism is devised to preprocess and formulate a series of CSPs via reasoning on the dynamically generated association map. The CSP solvers are interfaced to solve the formulated problems and feedback the results.

There are several software packages available, which avoid users being concerned with the constraint solving process. Several algorithms were employed; while these tools automatically determine a suitable algorithm or used the one selected by users. To name just a few, Numerica (Van Hentenryck et al., 1997) is one of the first constraint solving tools, which combines interval methods and several local techniques. It guarantees that all solutions are contained in the output boxes, even for nonlinear constraint satisfaction problems. 1stopt is a commercial software package, which is capable of solving complex nonlinear problems. In this solver, several algorithms were implemented, including simplex method, Newton method, Tabu search, etc.; and its own universal global algorithm is uniquely embedded (Cheng et al., 2009). Gecode (2015) is a free object-oriented constraint solver while supporting further programming to extend the capability. It has implemented a comprehensive set of constraints over discrete variables and several search engines and heuristics; interval methods were also implemented, but only a small portion of constraints, such as linear constraints, over float variables were supported.

2.5 Advances in Feature Technology

2.5.1 History of Feature Technology

The feature concept was initially inspired from the desire to support information integration between CAPP and CAD in the manufacturing field. A commonly accepted feature definition is "a generic shape associated with some engineering semantics" (Shah & Mäntylä, 1995). Due to the feature's pattern modeling capability, which can be easily associated with engineering concepts, especially for machining methods, features were demanded in digital manufacturing in the 1980s. Feature-based computer-aided manufacturing (CAM) systems offer flexibility for constructing conceptual geometry models, and are possible to automate the CNC code-generation process. However, in that early stage of CAD development, geometrical modelers were not feature-based. To transfer information from CAD to CAPP, machining features had to be recognized from the geometrical models. Extensive research has been conducted in feature recognition over the past three decades (Babic et al., 2008; Verma & Rajotia, 2010). The disadvantages of feature-recognition algorithms were their complexity and limited types of features that could be recognized (Lam & Wong, 2000). On the contrary, another mainstream technological approach is feature-based modeling, which builds models by using feature templates rather than recognizing features from an existing geometrical model. This approach contains rich information associated with design models (Xie et al, 2013). In this way, generic feature templates need to be predefined based on the identification of patterns with meaningful geometric and/or topologic characteristics to engineers (Shah, 1991).

Traditional feature technology in the mechanical design domain has already been well-established by many researchers, and the exploration of this research domain is not the focus of this research. For detailed information, please refer to Shah and Mäntylä (1995). In hindsight, it can be appreciated that traditional geometrical feature models were lack of flexibility in expressing engineering intents, and could only model feature information at a single level of abstraction. In theory, feature models can represent richer high-level information of products and can thus improve consideration of downstream engineering issues in the early design phase (Shah & Mäntylä, 1995). In later years, the concept of feature has been extended and used to bridge mechanical product design and engineering analysis, such as stress analysis with finite element method (FEM) (Deng et al., 2002; Lee, 2005, 2009) as well as manufacturability analysis (Cherng et al., 1998; Syaimak & Axinte, 2009). Multiple-view feature-based models have also been proposed to support the various stages of product development (Bronsvoort & Noort, 2004). Some non-geometrical relations were identified, but still, only geometrical relations were applied to connect different views. Considering the feature granularity aspect, features can be used to represent semantic information of different granularities from a hole, a single part, sub-assembly, to even a complex assembly (Chen et al., 2012; Ma et al., 2007; Xu et al., 2013).

Nowadays, feature-based modeling has been readily supported by most commercial CAD software tools, such as NX^{TM}, Dassault Catia^{TM}, and Solidworks^{TM}. As can be seen from the following subsection, feature technology is already capable of covering various phases of the product life cycle. In this research, the general definition of mechanical design features adopted is a set of generic geometric entities associated with some engineering semantics (Tang et al., 2013) with the extensions to many advanced feature types, including part features (Bidarra & Bronsvoort, 2000; Zheng et al., 2012) and assembly features (Ma et al., 2007).

2.5.2 Constraint Modeling in Feature-Based Modeling

In the feature-based modeling scheme, constraints are supposed to specify detailed associations among feature parameters within and across different features and are used to validate the consistency of features (Ma & Tong, 2003). Generally, constraints can be divided into two categories: geometrical constraints and engineering constraints/non-geometric ones (Anderl & Mendgen, 1996).

Geometrical constraint modeling has drawn great research effort; as a result, current commercial software packages can provide strong geometrical constraint modeling and solving capability. Bettig and Shah (2001) derive a standard schema for general geometric constraints to facilitate data exchange and development of neutral APIs to constraint solvers. Also, a comprehensive review of geometrical constraint solving has been addressed by Bettig and Hoffmann (2011), among which the graph constructive approach was recognized as a dominant solution for geometric constraint solving. However, geometric constraints are typically unary or pairwise relationships of geometric features (Van der Meiden & Bronsvoort, 2006). On the contrary, non-geometric constraints are usually not just simple 1-1 relationships. For example, the ASME code for pressure vessel specifies a constraint among the design variables pressure, thickness of vessel and allowable stress of material (ASME, 2004). In this case, the decomposition plans in the graph constructive approach have to be adjusted to accommodate engineering constraints.

Also, geometrical constraints only describe associations among engineering entities at the geometrical and topological level; functional relationships are still missing in the geometrical features. To effectively propagate engineering changes in a multi-disciplinary environment, both adequate information representation of individual discipline, as well as interactions and association management across disciplines are required (Bouikni et al., 2006), since such interdisciplinary associations cannot be represented by geometrical constraints but engineering constraints. Compared to geometrical ones, engineering constraints specify the associations across different phases/domains, such as mechanical and chemical process domains. Unfortunately, in current CAD software packages, only geometric constraints are stored in the solid models; non-geometric constraints, which can explicitly represent high level engineering semantics, such as design intents, are not specified explicitly yet. The examples of engineering constraints include design rules or codes, engineer preferences/assumptions, constitutive equations, inter-feature constraints across stages, etc. The variety of types of constraints imposes a major challenge of engineering constraint modeling. Besides, in most cases, engineering constraints cannot be predefined automatically during feature creation as the specific rule applied is hard to determine without engineers' interventions.

3 Feature-Related Terms Definitions

3.1 Introduction

To clarify the meaning of terms frequently used in this thesis, their definitions are specifically given in this chapter. The schema of chemical process features is defined based on the unified feature scheme, and hence it is introduced first along with its base concept, associative feature. Then, the definitions of the terms are as followed.

3.2 Associative Feature Concept and Unified Feature Scheme

As proposed by Ma and Tong (2003), a new feature modeling mechanism, associative feature, can handle the mutual constraints among engineering entities beyond the boundaries of geometric entities, such as solids and components, and enable change propagations with a consistent data structure. In their example implementation, the mold cooling channel subsystem can be updated coherently according to the change of a part that happened at a later stage of its life cycle. The concept of associative feature "bridges the gaps between the interfacing of knowledge-oriented tools and CAD applications, to aid intelligent product development" (Chandrasegaran et al., 2013). Further, Chen et al. (2006) extended the associative feature into a unified feature scheme in order to maintain the integrity and consistency of the product model and further enhanced the schema by sharing and managing dependency associations.

The unified feature scheme was based on a generic feature element, as is shown in figure 3.1, which provides common characteristics and embedded behaviors. Constraints are modeled as behaviors in order to validate feature models and hence maintain the consistency of the product model. Built upon the template of generic feature, the associative feature concept is a fundamental mechanism developed under this unified modeling schema; it can be applied to different engineering applications and stages, and thus supports information association modeling across the product life cycle. In this research, it is proposed to apply associative feature technology to the chemical process domain in order to provide unified feature definitions across chemical and mechanical engineering domains, and therefore supporting the collaboration among these domains.

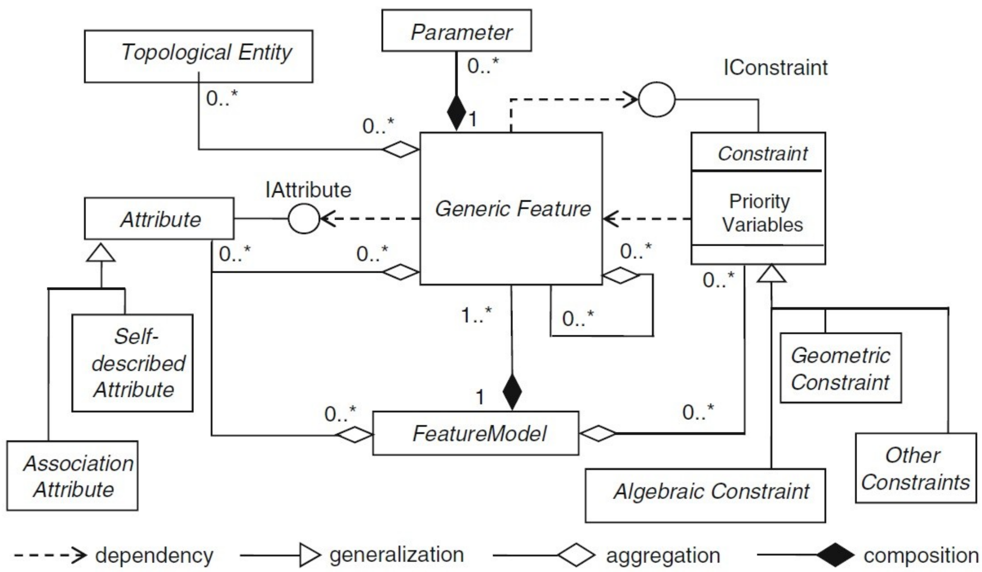


Figure 3.1 Updated semantic definition of generic feature definition (Tang et al., 2013)

3.3 Terminology Related to Feature

The definition of feature is based on the unified feature concept (Tang et al., 2013). A feature is defined here as an information unit which represents semantics or significance of a particular engineering interest. In the chemical process project context, the information that has to be explicitly represented and shared across disciplines is non-geometrical rather than geometric. Therefore, definitions of geometric features are out of scope of the research; instead, the parameters of geometric features refer to the parameters of high level mechanical features so that vertical reference relationship is established. A generic feature contains common characteristic and behaviors, e.g., association within one feature or across features.

Feature parameters are major constituents of a feature, which describe the characteristic information of the feature. The value of the feature parameter could be either fixed value or changed by engineers subject to some constraints. For example, thickness is one parameter of the pressure vessel feature; it could be of any value in the allowed set of values subject to the manufacturability constraint.

Constraints specify the relationship that has to be held among several feature parameters. The constraint refers to its governed feature parameters, and hence explicitly describes feature associations. The validity of feature has to be checked against the constraints which are imposed on the parameters of this feature.

In this thesis, a progressive expanded constraint satisfaction problem solving method is proposed to implement change propagation. Variables and constraints of the formulated constraint satisfaction problem are determined by searching the associated feature parameters and constraints within current scope. To share a common terminology with the constraint satisfaction community, variables, commonly used in the constraint satisfaction problem formulation, refer to feature parameters which are included in the scope of the formulated problem. The feature parameters involved collectively form the set of variables of those formulated constraint satisfaction problems.

4 Multi-Disciplinary Feature Definitions

4.1 Introduction

Currently, EPC companies are facing the challenges of reducing time-to-market due to increasing competition (Wiesner et al., 2011). However, it has been indicated by Bayer (Bayer & Marquardt, 2004) that design changes always exist and lead to the propagation across all the disciplines involved; however, these are time-consuming and error-prone in the current industrial practice. For example, design changes of the chemical process design will introduce the required changes of detailed equipment designs in the mechanical domain. On the other hand, the design changes of process equipment items need to be reconciled in the piping and instrument diagram (P&ID), and may also lead to major modifications of the P&ID. These kinds of associations need to be addressed and managed carefully in order to keep the information consistent. In addition, the knowledge needs to be represented in a way that it can be shared and interpreted by the engineers from different domains (Bañares-Alcántara, 1995; Bañares-Alcántara & King, 1997; Verhagen et al., 2012).

To facilitate the information sharing between chemical and mechanical engineers, the first step is to build a unified data model across these two disciplines. So far, the feature technology is primarily applied in the mechanical engineering domain. Based on the summarization of different definitions of features, Ma et al. highlighted two main characteristics of features: representing engineering semantics and being associated with a certain level of product information (Ma & Bong, 2010). They believed that features are capable of representing information in a hierarchical structure and at different granularities

(Chen et al., 2012) and also evolve along with different phases of the lifecycle (Bronsvoort & Noort, 2004). Such distinctive characteristics of features make the feature technology a perfect solution for the integrated collaborative platform. Hence, it is decided to introduce feature technology into chemical process engineering domain. In this chapter, a schema of chemical process features is developed first here to start the modeling of process engineering informatics entities.

4.2 Constituent Identification in Chemical Process Projects

The following terms are defined in order to establish a consistent terminology in the field of engineering informatics related to the chemical process engineering domain.

Function	a transformation of material streams from the input states to the output states that an expected chemical process is able to achieve
Elementary Process	a basic chemical reaction to convert from a set of input compounds to an expected output set
Unit Process	a process step that involves the "elementary process" of streams from a set of feed composition to the set of output streams with a basic unit of reacting vessel
Modular Process	a set of "unit processes" directly related to fulfilling the chemical conversion function from a given set of feed streams to another set of output streams with expected compositions with a self-supported operational module of commonly-recognized independency. Due to the generic nature of this concept, a "modular process" can contain another child "modular process."

	Therefore, there is a nesting mechanism for "modular process" to form a hierarchical processing system for a scalable plant or a project
Operational Process	an industrial scale process that produces expected chemical products and by-products via the systematic "chemical transformation and physical separation of materials" (Douglas, 1988)
Elementary Operation	the most basic physical (non-chemical) process within the practically scalable context that is deployed to change the physical state of the feed materials
Unit Operation	one or several associated "elementary operations," delivering a set of relatively clustered functions which are the processes of physical treatment of feed steams with a mechanical equipment unit, such as "preparing the reactants, separating and purifying the products, recycling unconverted reactants and controlling the energy transfer into or out of the chemical reactor" (McCabe et al., 2005)
Modular Operation	a set of unit operations to fulfill the physical states changing from a given feed stream to the output set of streams with expected states with reasonable independence in operational functions and self-supported equipment completion
Auxiliary Operation	a system consisting of a group organized physical operations which facilitate and support "operational processes"

Plant Operation a set of connected "operational processes" supported by "auxiliary operations" in a plant environment

Before starting to define chemical process features, the structure of a chemical process plant, different granularities of information from plant level to elementary level, as well as the relevant *attributes* and *methods* to characterize those entities, have to be identified (Shah & Mäntylä, 1995). It is assumed that a chemical process plant is the final result of engineering and design, which is to be put into operation after being committed. It is designed for a certain "plant operation," which can be considered as a combination of chemical processes and physical operations no matter how large the scale of the plant, as shown in figure 4.1.

The hierarchical relationship among the processes and operations mentioned above is shown in figure 4.1. As is shown on the left-hand side under the plant operation, *operational processes* are defined to convert the feed materials into the final products with by-products, and these conversion processes are supported by the "auxiliary operations." An "operational process" is, in most cases, composed of several "modular processes," as well as "modular operations," while a "modular process" will involve several "unit processes" and "unit operations." In each "unit process," there are one or several "elementary processes"; each of such "elementary process" represents typically explicit chemical reactions involved. Such reactions could be the expected or side reactions leading to the final products and by-products. Similarly, on the physical (non-chemical) process side, i.e. the right side of figure 4.1, the auxiliary operations are composed of "modular operations," which further involve one or more "unit operations." The "modular operations" could be steam-generation modules, utilities, etc. Note that "unit operations" are typically coupled with other “unit processes,” and used to support “modular process”; and similarly, “modular operations” can also be clustered into an “operational process” with other “modular processes.”

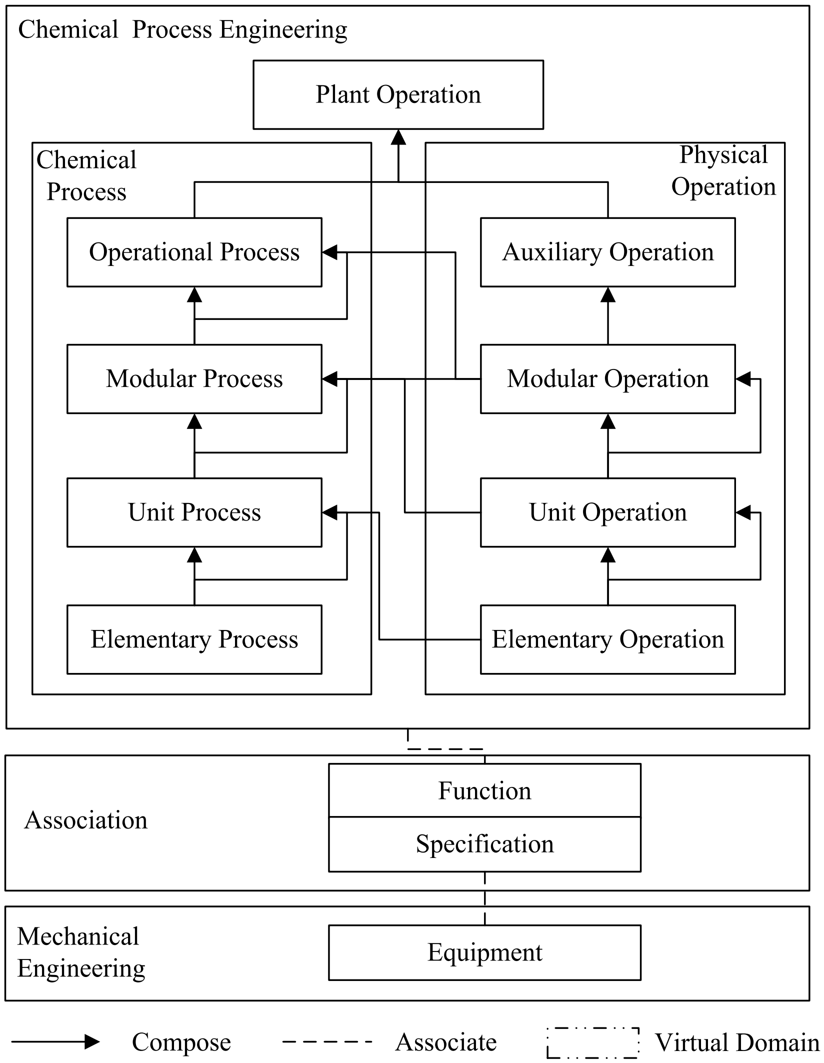


Figure 4.1 Structural relations diagram of a chemical process project

For some applications, it works well to list every chemical reaction involved in an “elementary process.” However, for some chemical process, such as refinery or oils sands upgrading, due to the complicated composition of streams, it is impossible to represent the chemical reaction each molecule undergoes (Chang et al., 2013). It requires the representation of the elementary chemical process to be scalable to describe the chemical reactions in coarse reaction combinations. In turn, a unit process consists of a set of elementary processes. For example, in oil sand upgrading operation, the hydrotreating process is considered as a unit process, which involves "elementary processes" such as aromatic saturation, hydrodesulferization, hydrodenitrogenation, etc. Note that in this case, the aromatic saturation is an elementary process containing a lump of reactions of all aromatics. Further, an oil sand upgrading operation has a primary upgrading and a secondary upgrading process; they are referred to as "modular processes." Each of them can be defined by many "unit processes," such as the hydrocracking and hydrotreating processes together with some other unit operations, such as pumping, preheating, and separation.

4.3 Associative Chemical Process Features

A set of associative chemical process features are proposed to represent the characteristic semantic attributes, relations, constraints and functions (in the form of *methods* in the object-oriented software modeling terminology) embedded in a relatively self-contained chemical process model. The proposed chemical process features resemble the structure of real chemical process engineering cognitive patterns shown in figure 4.1. The definitions of the associative chemical process features are defined as follows.

An *Associative Chemical Process Feature* is defined as a tuple of the sets of characteristic constituent entity *pointers, attributes* and *methods* related to a generic *class* representation for chemical processes. From chemical engineering application point of view, this class contains an abstracted representation of process flow defined in the terms according to processing semantics. Proposing this set of chemical process feature types in the chemical engineering domain in contrast to those typical features defined in the mechanical product domain is a key contribution of this research work.

The compulsory attributes include material balance parameters, e.g. volume flow rate, mole fractions of process streams; energy balance ones, e.g. heat exchanges and enthalpy changes; and process conditions, e.g. pressures, temperatures. Other than typical initialization, editing and deleting methods, validating methods for the flow balances are embedded in the class definition. The attribute set collectively expresses the parametric characteristics of those interested geometric entities and chemical process semantics, and the method is defined to provide functions for the feature model, such as *initiate()*, *modifiy()*, *validate()* and *delete()*.

The reason it is named "associative" is that it is built with "associative feature mechanism" (Ma & Tong, 2003) to manage associations among engineering entities. The chemical process feature is the parent class of the following eight features: "operational process features," "modular process features," "unit process features," "elementary process features," "auxiliary operation features," "modular operation features," "unit operation features," and "elementary operation features."

Elementary Process Feature is an abstracted phenomenon class incorporating engineering semantics characterizing the "elementary processes," such as chemical reactions. An instance of "elementary process feature" is given in figure 4.2 with the example of hydrodesulphurization and olefin hydrogenation. The engineering information, like reactants, outputs, selectivity, etc., is adopted to characterize the chemical reactions.

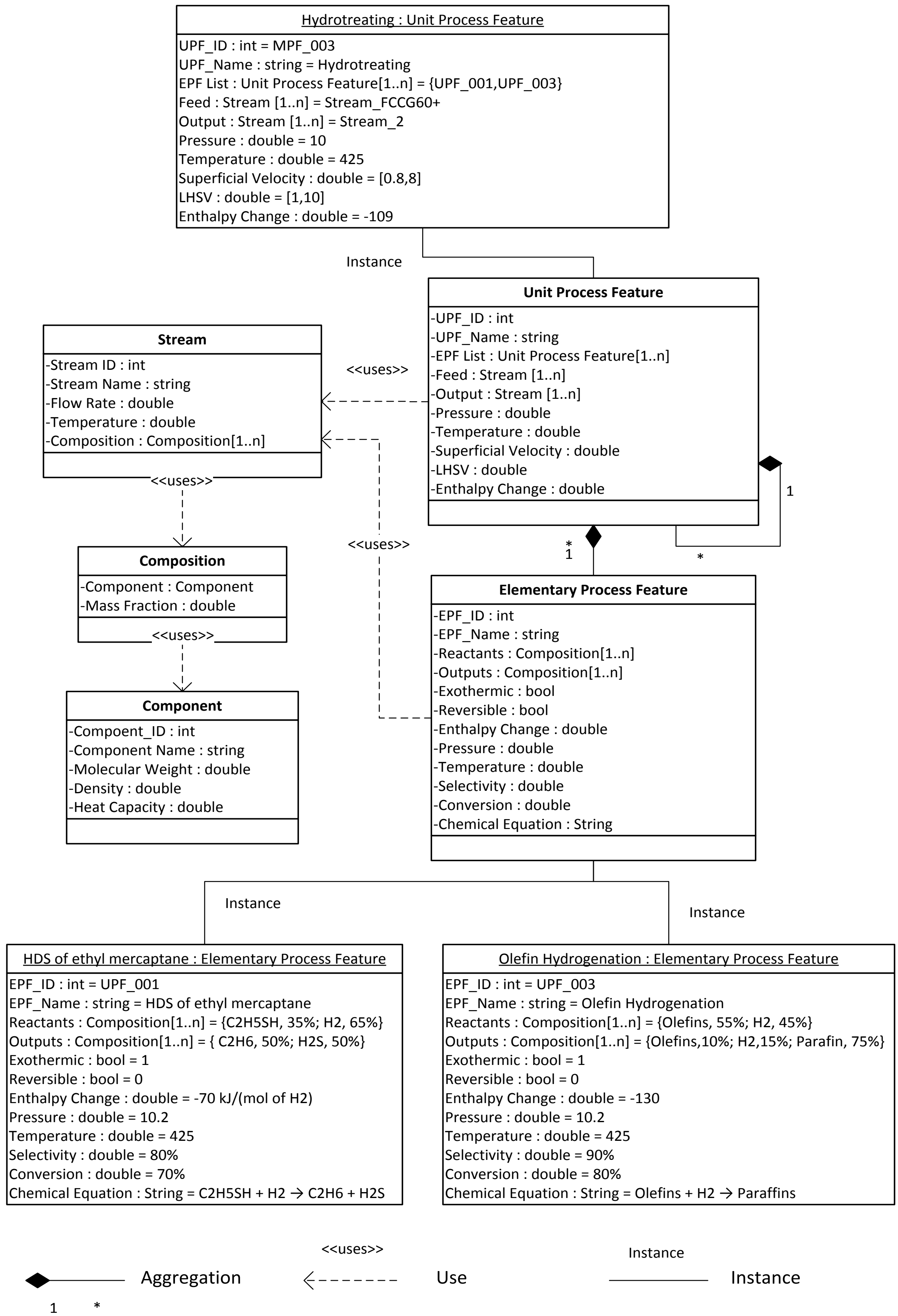


Figure 4.2 Relations among associative chemical process features and their instances

Unit Process Feature is an abstracted class associated with the process engineering semantic concept of “unit process” (usually corresponding to process flowsheet symbolic unit). The more detailed data structure of “unit process feature” is given in figure 4.2 with the example of a hydrotreating process. For example, a unit process feature involves attributes describing characteristics of a unit process in P&ID, such as those describing feed and output streams, temperatures and pressures. A unit process feature contains one or more elementary process features.

Figure 4.2 shows more detailed structures of the “unit process feature” and “elementary process feature” with example instances. The example of the unit process feature shows a typical chemical process in the current refinery industry, i.e. a hydrotreating process. As is explained, a unit process instance could include several elementary process feature instances, or reaction clusters. It could be represented either at the molecular level, as shown in the left instance of the elementary process feature, or in a lumped way, as shown in the right instance.

Modular Process Feature is an abstracted class representation of process flow circuit consisting of a number of unit process/operation features with engineering significance at “modular” level. This class has nesting capability. One of the necessary constituents is the flow circuit topology structure and it is also enabled with the connection relations or constraints of the member “unit processes,” “unit operations,” and even other lower level “modular processes.” Two example instances of this class can be the primary and secondary bitumen upgrading processes respectively.

Operational Process Feature is an abstracted class representation of "operational process" as discussed above, including the pointers of member processes and supporting functional entities, operational flowsheet, layout 2D or 3D models and block diagrams with operational level chemical process engineering significance. It mainly describes the operational level characteristics.

Elementary Operation Feature is a set of abstracted physical interactions with characteristics of the "*elementary operation*" embedded, such as the physical state changes of the streams of the "elementary operation."

Unit Operation Feature is an abstracted class associated with the engineering semantics of the "unit operation" (usually corresponding to process flowsheet symbolic unit).

Modular Operation Feature is an abstracted representation of process flow circuit of the "modular operation" with engineering significance. Similar to "modular process feature," it also involves topology and connection information of composed "unit operations."

Auxiliary Operation Feature is an abstracted shape or block diagrams of the "auxiliary operation" with engineering significance.

The proposed set of process features is to provide an abstracted model of chemical process semantics with different granularities. Note that the constituents of a feature of interest are different at different levels of granularities. For example, the modular process features interest topological information of the process flow diagram; while unit process features emphasize on the operating conditions of each unit.

It is worthwhile to note that there are a number of complicated reference relationships among the constitutional chemical processes and physical operations. An implicit example is that the outlet flow rate of the upstream equipment items determines the flow rate of downstream ones. Another example is the association between heat exchange operation and reaction process. The operating temperature and flow rate of the reaction determines the flow rate of the heat exchange operation and may even affect the internal layout pattern of the heat exchanger, which supports the exchange process. These kinds of associations listed above are all defined in the "modular process feature." Sometimes, one unit process feature may even determine the existence of another unit process/operation. For example, the operating pressure of the reaction process may drop to a limit where the pumping operation is no longer needed.

4.4 Mechanical Features

In the chemical process project context, the mechanical features describe the mechanical engineering information of the process equipment. Take the reactor as an example; the compositions as well as the associations among the components can be described by the assembly feature, while the detailed engineering information, e.g., thickness of pressure vessels, is embedded in the part features. These mechanical features will further drive the update to the geometrical features in the mechanical CAD systems; the CAD models are supposed to be built in a parametrical way.

4.5 Summary

A schema of chemical process features is developed in this chapter based on semantic explorations of the chemical process engineering domain. The associative chemical process features provide comprehensive and flexible definitions of chemical processes in a hierarchical structure. Related to those existing product-related features in the mechanical engineering domain, the associations across multiple domains can then be managed through the features. How to manage such interdisciplinary associations is further discussed in chapter 4.

5 Unified Interdisciplinary Engineering Framework

5.1 Introduction

Although chemical and mechanical engineering disciplines have their own problem-solving focuses and specific interests, they share some common characteristics in product and process semantics. This chapter starts with finding commonalities between chemical and mechanical engineering domains. With those commonalities, it is beneficial to build a unified interdisciplinary engineering framework. Under this framework, the inter-domain function feature is developed to manage functional association between chemical process features and mechanical features. Such a framework is believed to be very helpful in enhancing the interactions across the disciplines.

5.2 Plant Development Observations

Developing chemical process plants and developing mechanical plants share similar generic processes, which involve engineering, procurement, construction, and production phases. In each phase, they involve different instantiated activities, and in some cases, are only different in name, as shown in figure 5.1.

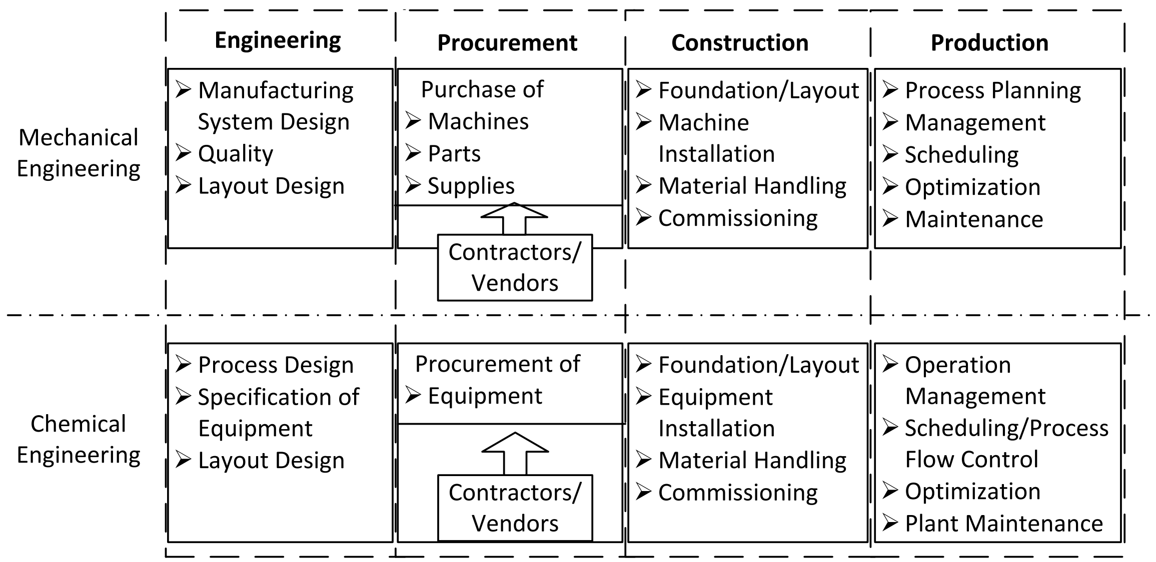


Figure 5.1 Process commonality in mechanical and chemical plant developments

For example, in the plant engineering phase, the core effort in the mechanical domain is to design the production processes, as well as their equipment and material handling space layouts, namely the modeling of the manufacturing system. In chemical engineering, there exists similar chemical process modeling. Certain differences are observed in engineering contents, such as the interest in quality of products. In mechanical engineering, it focuses on tolerances in dimension and surface roughness, as compared to purity or composition within chemical engineering. However, both sets of quality attributes are designed to achieve a production quality with the required capacity.

In the procurement phase, a mechanical manufacturing system involves the purchase of specific machines, parts as well as other supplies, while the procurement of chemical production equipment involves the purchase of pumps, heat exchangers, etc. In the construction phase, both fields work on the construction of foundation/layout, the installation of equipment/machines/production lines, and material handling systems according to engineering design.

One major difference in the production phase is that a mechanical plant involves process planning. This is due to the large variety of products that a mechanical plant manufactures. Especially for some manufacturing job shops, they produce high-mix, low-volume products. In contrast, large-scale production is commonly involved in a chemical process plant. Such flow production is very similar to large-scale assembly lines in manufacturing where the processes are more or less fixed and the production is running continuously with the feed of raw materials, much like a continuous chemical process plant. In both domains, some aspects, such as the management, scheduling, optimization of production, and maintenance of equipment/machines, can be observed.

It is worthwhile to note that, like in mechanical manufacturing, information among the chemical process engineering, procurement, and construction (EPC) activities should be well-modeled and kept consistent across the phases in order to achieve the sustainability of the whole project.

5.3 Integration of Product and Process Models in Engineering

Products and processes are two sets of interrelated engineering informatics models, which are commonly developed in any engineering domain, such as chemical engineering and mechanical engineering. For example, a final product model represents a finished good that meets the desired functional requirement. In contrast, a process model describes a designed procedure/method used to produce the whole or a part of the product. As shown in figure 5.2, the design of a product demands process design and engineering, while the production process, implemented from the process design, creates the product or several products from raw materials. Such interdependency exists in all engineering domains. For example, in the chemical engineering domain, a chemical process creates products, such as synthetic crude oil, petro-chemicals, pharmaceuticals, etc., from various raw materials. It can even be argued that chemical products and mechanical products can share a similar product design methodology (Moggridge & Cussler, 2010). The above discussion inspires a unified engineering model to support interdisciplinary interactions. The reason for this is that a chemical engineering project involves both a chemical product and a chemical process in the chemical engineering domain, as well as a mechanical product and process in the mechanical engineering domain.

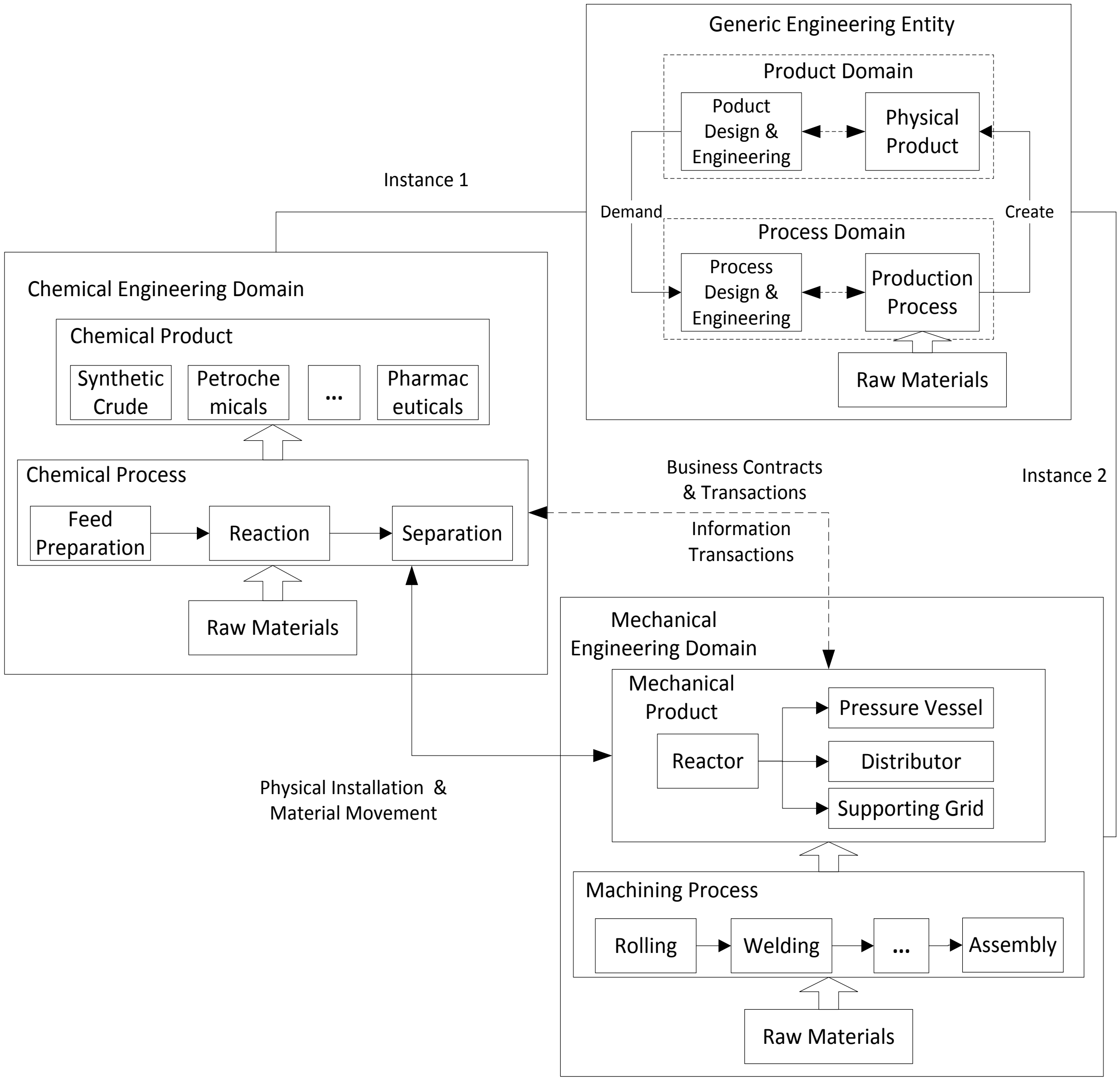


Figure 5.2 Generic information abstraction of engineering domains with commonality

Besides the interrelations between the products and processes in individual domains, there are also associations across different domains, such as those between chemical processes and mechanical products. The chemical process designed depends on the equipment items required to operate it, which are the mechanical products, such as reactors, heat exchangers, etc. They are designed to support the chemical process operation. Partially listed relations between the domains of chemical processes and mechanical products are demonstrated in figure 5.3. To model an interdisciplinary and unified engineering environment for chemical engineering projects, a set of comprehensive features are defined and the associations among these features are discussed in the next section. Among them, inter-domain functional feature is proposed to manage such functional mapping.

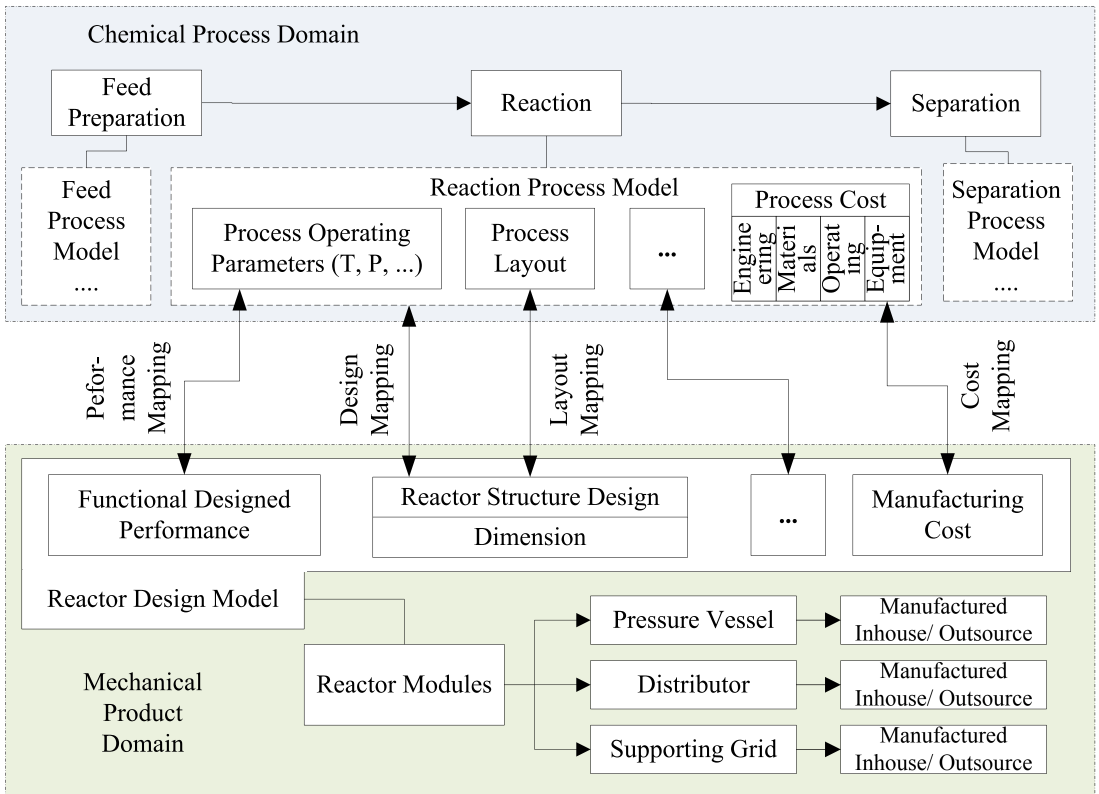


Figure 5.3 Partially relations between chemical process and mechanical product

5.4 Inter-Domain Functional Features

As is analyzed in section 4.2, chemical functions should be mapped to mechanical functions provided by the mechanical devices in order to complete an engineering project cycle. An associative feature is believed to be capable of representing a set of relations (Ma & Tong, 2003), configurations, and functions (Brown, 2002; Bronsvoort & Jansen, 1993). The authors attempt to use associative features to handle the layered representation of functions, as well as the relations among them. Here, as shown in figure 5.4, an inter-domain functional feature class is developed based on associative feature definition which maps one domain's functions in the form of feature set to another's. In the inter-domain functional feature class, two function representation schemes are combined as described in Erden et al. (2008) to associate the function of a process and the process equipment. That is, a function is represented as a feature set consisting of "verb + noun" pairs as well as the corresponding "input and output" flow transformations with governing specs and constraints. In this research, for example, an inter-domain functional feature maps functions required by chemical process with mechanical functions provided by a set of mechanical equipment. Mechanical functions are further associated or achieved by a set of mechanical functional features as in Shah and Mäntylä (1995) that it represents "sets of features related to specific function, which may include design intent and non-geometric parameters related to function and performance." The functional feature is defined to support function representations as well as its supporting mechanical features and associated performance parameters.

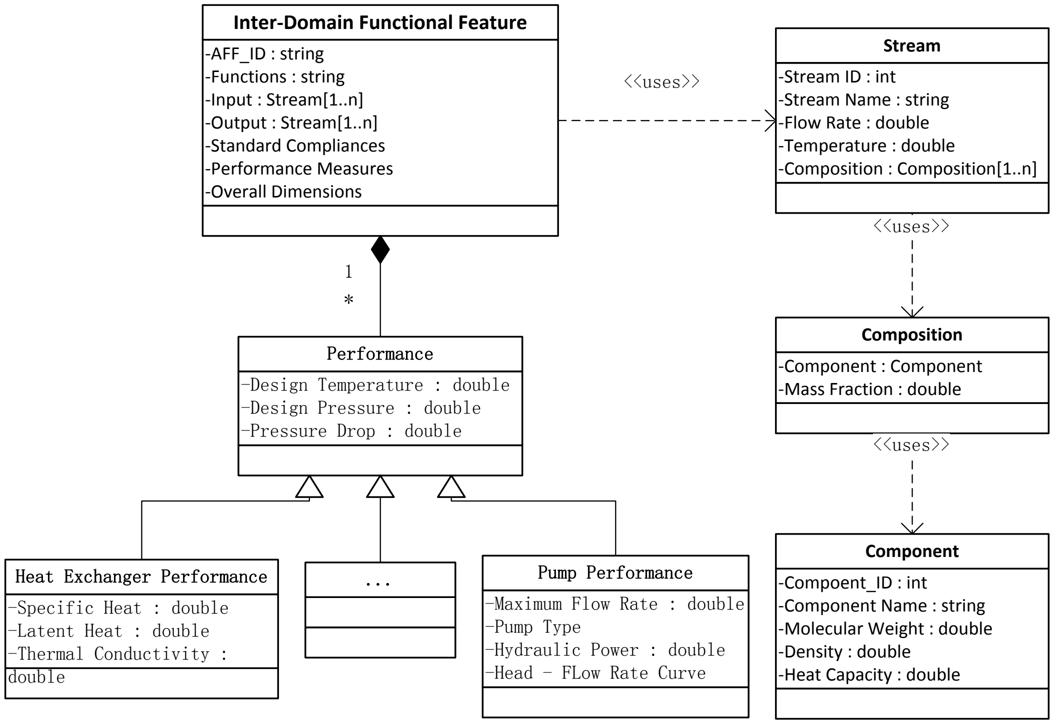


Figure 5.4 Functional feature definition

The question of how to explicitly represent functions is crucial, as they collectively specify the design intent from an inter-disciplinary engineering perspective. In chemical process engineering, the goal is to create a system that performs a series of chemical production functions to produce certain products with the capacity requirement. Commonly, the design efforts consist of many domain-specific engineering tasks to realize functions by using function features. For example, one task is to design a chemical process to perform those chemical functions, while the other is to design equipment that performs the expected system functions in support of those chemical production functions. Therefore, functions can only be achieved by working closely in association with different disciplinary function features, such as those between chemical process engineering and mechanical engineering. This is exactly what purpose the inter-domain functional feature serves.

There are nine process-level mechanical function primitives listed in table 5.1. The attribute values of one function feature could trigger the initiation of another functional feature. For example, the properties of the outlet of the reactor and the required output could indicate that there is a need for the function "ToSeparateFlow," while the difference between the inlet pressure, feed, and elevation implies the function "ToDriveFlow." The associations between the chemical process feature and the inter-domain functional feature provide the attribute values of functions, and hence get different function instances. Taking "ToDriveFlow" function as an example, figure 5.5 illustrates the inheritance relationship. It has two child classes, i.e. "ToDriveLiquidFlow" and "ToDriveGasFlow." Different parameters are used to show different function instances, as shown in the bottom level of figure 5.5.

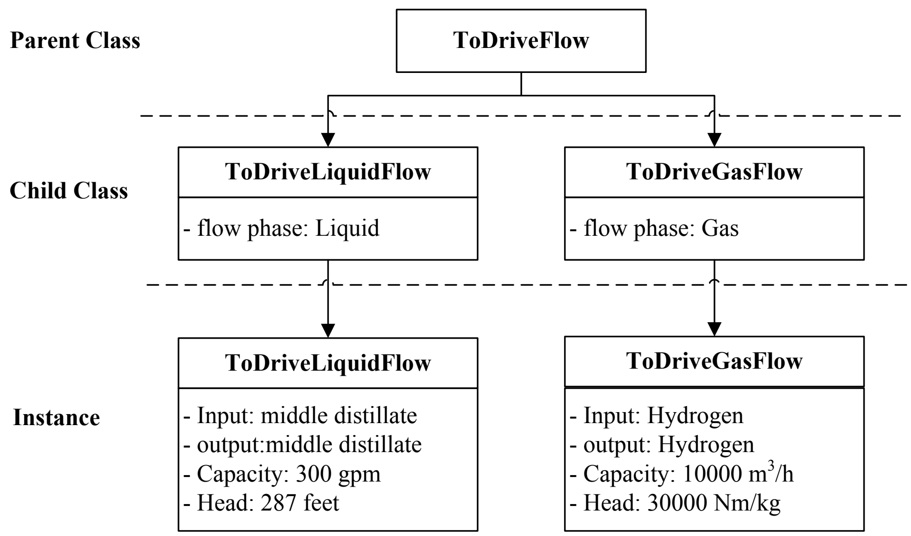


Figure 5.5 A case of inheritance relationship of the "ToDriveFlow" function class

These process-level functions will be further broken down into sub-functions, which specify the functions of components of the equipment. An example of a fixed-bed reactor is shown in table 5.2. The "ToReactFlow" function is supported by six sub-level functions.

Table 5.1 Process-level function primitives and functional taxonomy of equipment (Siirola et al., 1971; Mahalec & Motard, 1977).

High-Level Function	Functional Devices
ToSeparateFlow	Separator
ToReactFlow	Reactor
ToMixFlow	Mixer
ToSplitFlow	Splitter
ToTransportFlow	Pipeline
ToDriveFlow	Pump
	Compressor
ToTransferHeattoFlow	HeatExchanger
ToStoreFlow	StorageTank
ToControlFlow	Valve

Table 5.2 Sub-level functions of the “ToReactFlow” function.

	Sub-Function	**Functional Components**
F2.1	ToSupport	Pressure Vessel
F2.2	ToFeedinFlow	Inlet Diffuser
F2.3	ToDistributeFlow	Distributor
F2.4	ToSupportCatalyst	Supporting Plate
F2.5	ToQuenchFlow	Mixing Tray & Quench Pipe
F2.6	ToCollectFlow	Outlet Collector

One of the methods defined in the inter-domain functional feature class is to achieve the maximum mapping from chemical functions to mechanical ones. The required functions have “m-to-n” reference associations with process features, i.e. the parameters of required function change along with the process feature attributes. Similarly, the provided functions also have “m-to-n” reference associations with mechanical features. Specific constraints are used to express detailed associations between those variables. Functional association constructions can be classified into three categories: preliminary mapping, interactive configuration and customized design, as is shown in figure 5.6.

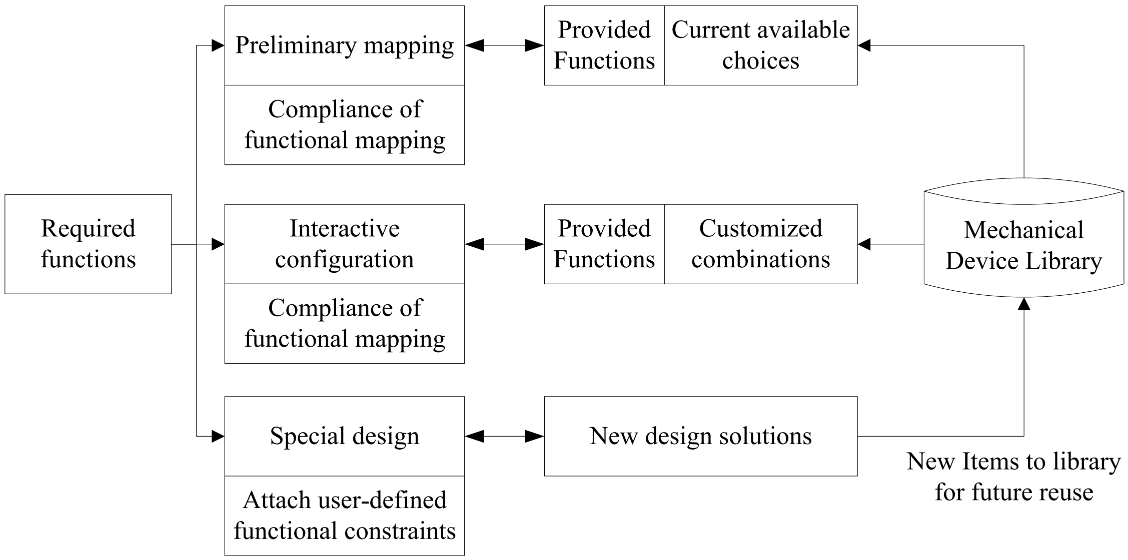


Figure 5.6 Inter-domain functional mapping mechanism

Preliminary mapping is a 'handshaking' mapping mechanism proposed to manage the associations between required function and the functions provided by those off-the-shelf equipment items supplied, such as those existing standardized valves and pumps. Based on the functional taxonomy of process equipment, such standard units are built into the equipment library. Along with this, the functions provided by the equipment are represented with generic functions and further performance details. The performance specifications of each equipment item are dynamically calculated when being considered based on the operating conditions. Functional compliance between the required functions and provided functions is checked on two levels. First, the general function types are checked. Then, the required functions are compared with provided function in detailed performance specifications. Based on the specified constraints and evaluation metrics, the compliance index can be calculated. The recommended equipment items are offered to engineers according to the compliance index.

Interactive configuration is an intermediate mapping mechanism between preliminary mapping and customizable solutions. A similar mapping mechanism is applied. The difference is that a set of configurable equipment items, instead of one single equipment module, are structured to map required chemical process functions. For example, two pumps in parallel or serial connections to provide the required pressuring function, which is to drive the liquid to a certain head in a specified flow rate. Considering the combinatory complexity, this mapping mechanism still needs engineer intervention. Intelligent algorithms can be further developed to such support configuration design and engineering.

Special design is an in-depth cross-domain function mapping with full interactions defined. Key equipment items in a process, such as reactors, distillation towers and pressure vessels, are to be specially designed. In this scenario, the required functions of chemical processes 'pull' the design solution of process equipment. The constraints involved in the design are described as the associations between process features and mechanical features. For example, the ASME code is applied to calculate the thickness of the pressure vessel based on the design pressure; while the design pressure is usually set 10% higher than maximum operating pressure of the specific process (Couper et al., 2010). Generally, the constraints are expressed as tuples C={V, E}, in which V is a set of feature variables associated with the constraint, and E is the expression of the constraints with the global index of each variable. All the user-defined functional constraints are added to facilitate future validation in case of a design change happens. It is to be noted that several functions could also be mapped to one equipment item in this category. For example, a reactor can be designed to map both reaction and separation function required by reactivedistillation process. The newly designed mechanical equipment along with those specified constraints will be stored in a repository of knowledge for further reuse or reference as a past case.

5.5 Plant Layout Features

The layout design is highly dependent upon the target site conditions, including the atmospheric conditions, the available area, and the geology of the site (Bausbacher & Hunt, 1993). It is also closely associated with process design and equipment design, and subjects to engineering modifications (Schmidt-Traub et al., 1998; Persson et al., 2009). Plant layout can be modeled as a set of associative features with a systematic approach. For example, the dimension attributes of the plant structure layout and the space to accommodate each equipment item can be the class attributes that are associated with the mechanical design feature. Such a layout design feature needs to comply with the process flow circuit and safety considerations, as well as the connections specified in the process design feature (Guirardello & Swaney, 2005).

The plant layout feature can follow common rules extracted from the design code database to check the layout validation. Design changes can be imposed by the associated design feature methods, such as the safety distances between the equipment items and avoidance of interference for equipment accessibility for installation, upgrading, and maintenance. Depending on the evaluation procedure and conditional options, engineering changes related to the plant layout features may be just a dimensional update following the changes of equipment, or they may lead to total structural changes. For example, when the required layout change is due to certain space interference with other equipment items, or violation of rigid rules, such as a safety distance, structure redesign of the plant layout then becomes necessary.

5.6 Inter-feature Associations in Chemical Process Projects

Based on the identification of structural relations existing in a chemical process project, the chemical process features are proposed to represent the semantics embedded in the chemical process domain, in contrast with the typical feature models in the mechanical domain. The chemical process features are defined in a hierarchical structure to comprehensively model the system of chemical processes. Besides, the inter-domain functional feature is proposed here to manage the associations between the chemical process features and the mechanical design features. It associates the functional requirements specified in the chemical process domain with the functions provided by mechanical design feature developed in the mechanical engineering domain. Each feature class is associated with the "built-in" methods of creating, editing, and deleting the instances of features, which are omitted from figure 5.7 for clarity.

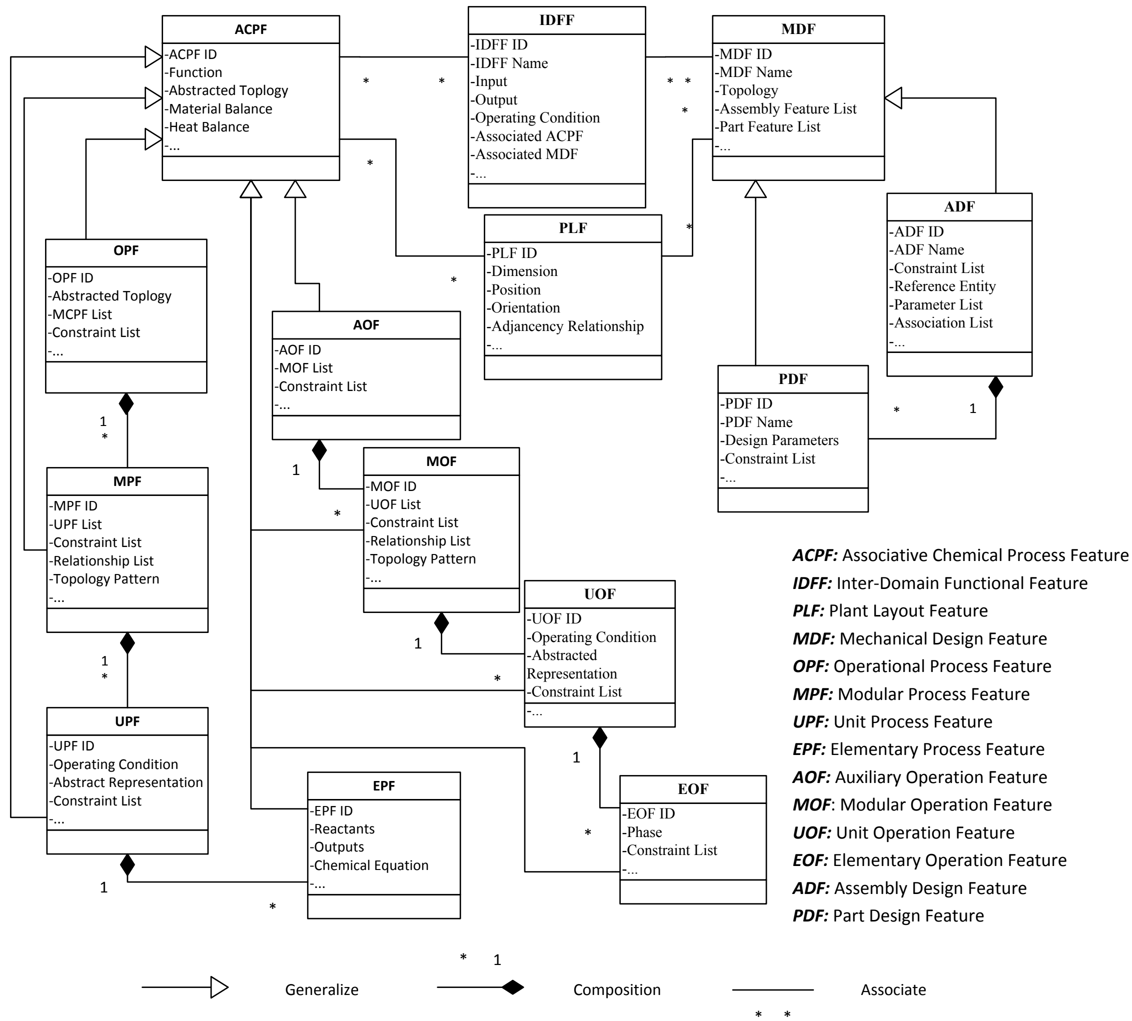


Figure 5.7 Relations among associative chemical process features and their instances

The associations established among those domain-specific feature definitions will benefit the information consistency in implementation of the integrated engineering environment and can lead to automatic design change propagation to address the inconsistencies. Whenever there is a design change, either in the chemical process design or the mechanical design, a validation mechanism is invoked, and two situations may occur. First, the design change passes the validation. This means that the design change has no conflict with any other design intent. The other possibility is that it conflicts with other design features, and thus, the change has to be propagated to resolve the design conflicts. The details of feature validation and change propagation are further addressed in the next two chapters. In the case of no feasible solutions, the associative changes to other design features are published and sent to the engineers who are responsible for the relevant issues. A compromise has to be made among the engineers.

5.7 Summary

This chapter proposes a complete framework for unified interdisciplinary engineering to cover both chemical and mechanical domains based on the advanced feature technology. The proposed inter-domain functional features build a bridge to associate chemical process features and mechanical design features, and therefore the design intent in chemical engineering can be expressed more explicitly in a tangible object form. Such functional features benefit the synchronization between the chemical and mechanical engineering domains. Applying inter-domain functional features allows for the establishment of flexible associative relationships among interdisciplinary features, which supports convenient updates of functional mapping.

The innovation of this proposed framework is that the dependency and references among different domain features are automatically well maintained by the system, and hence the cross-checking of inter-domain constraints is carried out in the background without users' intervention. The easy access to inter-domain data will save the communication hassle for engineers to obtain the precise and contextual information as required. Therefore, the information load for individual users can be less than before. Hence, the inter-domain engineering consistency can be maintained without overwhelming effort. The detailed modeling and systematical management of interdisciplinary constraint associations are further illustrated in the next chapter.

6 Interdisciplinary Constraint Association Modeling

6.1 Introduction

Collectively, engineering intent consists of many associative relations among engineering design entities; it has to be maintained throughout a product life cycle. However, in the current industrial practice, the lack of explicit representations and management of such associative relations prevents the engineering intent management and sharing across computer-aided systems in different phases of life cycle, and hence the collaboration among engineers cannot be fully facilitated. Since the complexity of projects has increased exponentially in the past decades, there commonly exist tremendous complex associations in a chemical process design. Even experienced engineers cannot always visualize the various associations among tens of thousands of engineering entities, especially those interdisciplinary associations. Modeling such complex associations is necessary to many engineering activities, e.g., identifying impacts across the disciplines in the process of implementing the change propagation. So far, the evaluation of engineering change impacts cannot be done systematically and quantitatively due to the lack of an explicit representation of associations among the entities in a complex engineering project.

Engineering constraints explicitly specify non-geometric-relations that have to be held among a set of variables along the lifecycle (Xue & Yang, 2004). With a feature-based representation established for the common characteristic information of engineering entities that exist in engineering design of chemical process projects, the constraints are used here to provide detailed representation of associations reflecting the engineering intent of engineers. However, methods to represent validate and maintain such non-geometric constraints are insufficient (Ma et al., 2008). The objective of this chapter is to propose a consistent set of engineering constraints to model inter- and intra-feature associations, especially interdisciplinary ones, as well as a set of generic mechanisms for the unified constraint model. Starting with summarizing commonly existing non-geometrical constraints in engineering design of chemical process projects in section 5.2, details on modeling and construction of such constraints association are given in detail in sections 5.3 and 5.4, respectively. In section 5.5, design model validation criteria are introduced. With systematic management of feature models and constraints, a developed mechanism to dynamically generate a feature parameter association map, is illustrated in section 5.6.

6.2 Engineering Constraints in Chemical Process Projects

The author is interested in chemical process projects due to the local industrial demands. The following types of non-geometrical constraints that commonly exist in the engineering design of chemical process projects are considered.

- Realistic Engineering Constraints

Realistic engineering constraints restrict the values of respective feature parameters, which could also be imposed by other disciplines. In most cases, such constraints are imposed based on downstream engineering considerations, such as manufacturability and shipment of the equipment from vendors to field. For instance, the maximum thickness of a pressure vessel is determined by the capability of available rolling machines; engineers may also impose constraints on maximum allowable thickness due to economic considerations. Another example is that shipment of equipment will limit the external dimensions of a pressure vessel if the shipment has to go through a certain bridge. In that case, the external diameter of the equipment cannot exceed the width of the bridge.

- Mechanistic Models

Mechanistic models are the mathematical equations derived from first-principles, such as conservation of mass, momentum or energy, which describe the behaviors of the targeted system. The equations specify the parameter relationships among their governed variables. It needs to be noted that the mechanistic models are usually derived based on a series of assumptions and simplifications, which would limit the application to only a certain range of cases. Such implicit applicable conditions also need to be modeled in the constraints when mechanistic models are built in the constraints.

- Data-Driven Models

In reality, mechanistic models are not always available, and sometimes the data-driven models (Gernaey & Gani, 2010) are employed, which represent the relationships of associated engineering entities by a set of data matrix or its deriving equations. Such data driven models could for example be obtained from specific experiments and simulation or derived from the data collected from an existing plant. One example is the performance curve of a pump. The associations among output head, volumetric flow rate, efficiency, and impeller diameter are specified in the performance curve. Change of any parameter due to process functional requirement change, would lead to the change of output of the pump according to the performance curve. For example, the change in the flow rate would lead to the change of output of head and pump efficiency. Sometimes the flow rate falls out the bounds of acceptable efficiency range, so a new pump or an impeller of another size is considered to meet the functional requirement.

- Connectivity

Connectivity is one common existing relationship in chemical process systems, which specifies how the unit processes and operations collectively compose the whole system. However, the state of the connectivity determines the existence of other constraints, which could be in other disciplines. For example, whenever two unit processes connect with each other, the mass balance constraint has to be held, specifying that the "operating flow rates" of these two unit processes are equal. The attributes of both units, such as flow rate and flow type of the corresponding process features, are required to be compatible with each other. Such connectivity relationships also need to be passed to layout features and mechanical features of corresponding process equipment items. The parameters, such as nominal diameters of fittings of connected equipment, have to be kept consistent as specified in the mechanical assembly feature.

- Design Codes or Standards

A set of engineering standards has to be followed when doing the engineering design. Such standards should also be modeled as the constraints in the feature models to assist easy identification of violations when the values of associated parameters are changed either by engineers or propagation from other changes. Due to the location of the project, the design codes employed may vary. One commonly used code in North America is the ASME (2004) code. Note that the specific conditions that determine the applicability of design codes need to be modeled as well.

- Design Decisions

There are extensive design decisions to be made along the life cycle of chemical process design. Whenever a design decision is made, the possible assignment to certain variables will be reduced. This also needs to be modeled in this system as a constraint to the respective feature parameter, either as a fixed-value constraint or a bound constraint. Such design decision constraint could fall into either soft constraint or classical constraint category, depending on the strength specified by engineers.

6.3 Systematic Constraint Management

6.3.1 Unified Constraint Model

Engineering associations are proposed to be explicitly expressed by constraints in this feature-based modeling scheme. To support modeling associations among features, generally, the schema of a constraint C is modeled as a tuple of governed variable set, expression and strength index, in which the variable set is a composition of pointers to those specific feature parameters. Formally C = {GV, Exp, SI}. GV is the variable set, which is the set of feature parameters governed by this constraint; Exp is the expression of the constraint, which explicitly specifies the detailed association relationships among the govern variables with their identity index; SI is the index showing the strength of this constraint. The associations among constraints, features and feature parameters are shown in figure 6.1.

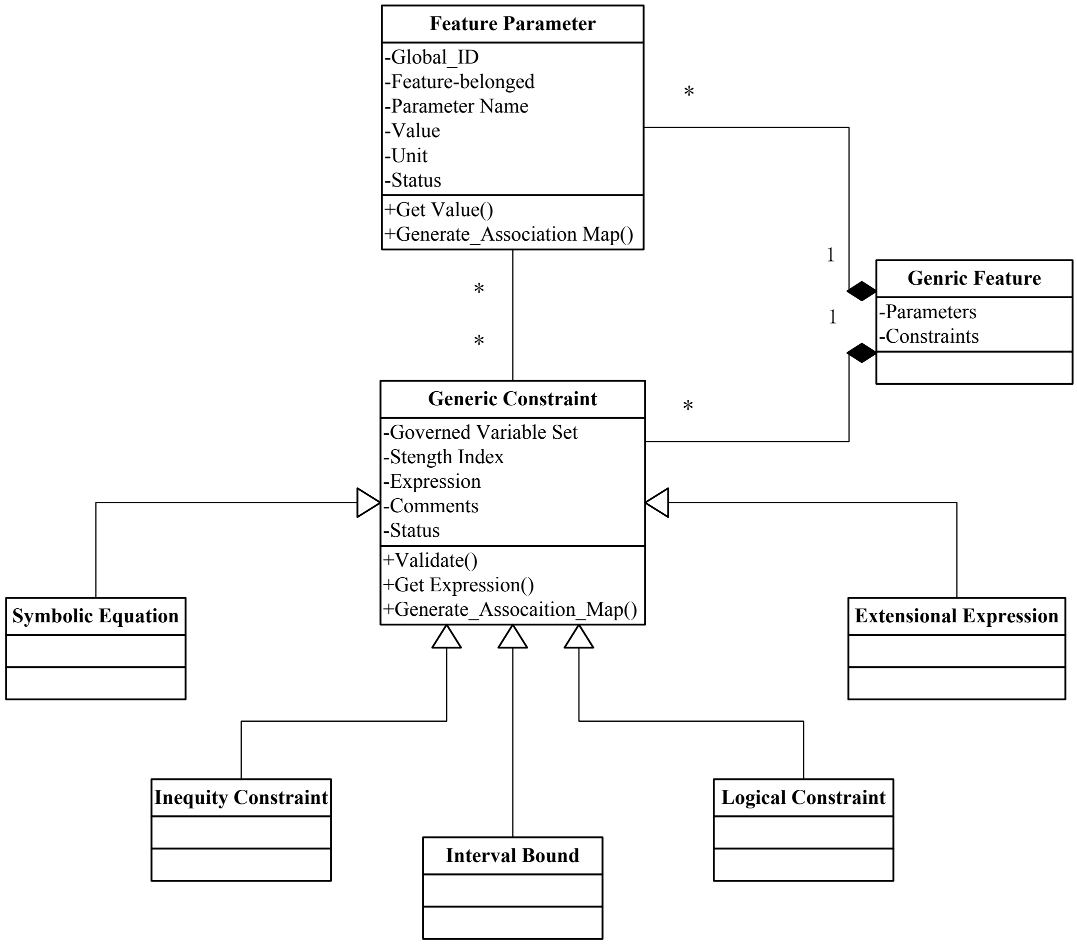


Figure 6.1 Unified Constraint Model

Constraints can be dynamically added by domain engineers to specify the above-mentioned associations among 1 to n feature parameters. Comments can also be added to illustrate the reason why this constraint is instantiated. This could facilitate the information sharing with other disciplines and future reference. Note that the feature parameters are globally indexed to facilitate constraint specification. The constraint expressions are symbolically represented with the global index, which could be in different forms depending on the types of constraints, such as symbolic equation, inequalities, logical expression or explicitly specifying a set of permitted tuples. As is shown in figure 6.1, different constraints inherit from the generic constraint and may override generic methods.

6.3.1.1 Symbolic Equation Constraint Expression

Symbolic equations are used to express any relationships among feature parameters that can be formulated in the mathematical equations. These are the most common constraint expressions, which could be used to represent some of design codes as well as mechanistic and data-driven models.

6.3.1.2 Inequity Constraint Expression

The inequity is another type of symbolic expression. In contrast to equations that describe equal relationships, this type of expression is for unequal ones. It is commonly used to express some design codes or some rules of thumb.

6.3.1.3 Logic Constraint Expression

A logic expression is used to express complex relationships among variables by specifying relationships among constraints, which mainly contributes to the modeling of design codes / standards.

Two common logic expressions are defined as follows.

Definition 1: If (expression_1) Then (expression_2)

Example: IF ($T > 650$ & $T < 750$) then ($S = 15650$), in which T stands for temperature and S stands for material strength. This specifies the logic relationships between temperature and material strength.

Definition 2: Either (expression_1) or (expression_ 2)

Example: either (DP > 1.1 * MOP) or (DP > MOP + 25), in which DP stands for design pressure of the vessel, MOP is for the maximum operating pressure. The constraints states that the design pressure should be either 10% or 25 psi higher than the maximum operating pressure (Couper et al., 2010).

6.3.1.4 Interval Bound

Interval bound is a unary constraint specifying the upper bound and lower bound of the variables, especially for continuous variables. Since this is designed for engineering design, there exist no negative values. Therefore, the default lower bound is 0 in general.

The interval bound is defined as follows:

IB (feature_parameter_index, strength index, (lower bound, upper bound))

6.3.1.5 Extensional Constraint Expression

Some constraint relationships among feature parameters can only be specified extensionally, i.e., in the form of set of satisfying tuples of its governed variables. As is shown in table 6.1, the design rule which specifies the relationships among the variables of number of pass, tube type, tube outside diameter, pitch, tube layout pattern, number of tubes and internal diameter of shells. In this example, the constraints in the extensional expression will be tube_count_relation (p1, p2, p3, p4, p5, p6, p7) = {(1, F, ¾, 1, T, 33, 8), (1, F, ¾, 1, T, 57, 10)…, (1, F, ¾, 1, S, 53, 10), …, (2, U, 1, 1¼, S, 58, 15¼)}, in which p1, p2, p3, p4, p5, p6, p7 are number of pass, tube type, tube outside diameter, pitch, tube layout pattern, number of tubes and internal diameter of shells, respectively; while F stands for Fixed tube, U for U-tube, S for square pattern, and T for triangle pattern. It needs to be noted that this type of expression only works for finite domain variables.

Table 6.1 A partial heat exchange tube sheet layout count table (Couper et al., 2010)

Pitch \ I.D. of shell (inch)			8	10	12	13¼	15¼
One-Pass	Fixed Tube	¾" on 1" △	33	57	91	117	157
		¾" on 1"□	33	53	85	101	139
Two-Pass	Fixed Tube	¾" on 1" △	28	56	90	110	154
		¾" on 1"□	26	48	78	94	126
		1" on 1¼"□	12	26	40	56	76
	U-Tube	¾" on 1" △	8	26	60	72	108
		¾" on 1"□	12	30	52	72	100
		1" on 1¼"□	X	12	22	38	58

6.3.2 Modeling of Engineering Constraints

The common types of engineering constraints have been discussed in section 5.2. Depending on the data available, one type of constraints could fall into different expression forms as stated above. For example, dimension limits could be expressed in the extensional way or the form of interval bounds and inequity expression. The possible expressions for all categorized engineering constraints are listed in table 6.2.

Table 6.2 Modeling of engineering constraints

Engineering Constraints	**Expression Types**
Realistic engineering constraints	Extensional expression Interval bounds Inequity expression
Mechanistic models	Symbolic equation expression

Data driven models	Symbolic equation expression Extensional expression
Connectivity	Logic expression
Design codes / standards	Symbolic equation expression Inequity expression Logic expression
Design decisions	Extensional expression Interval bounds

6.4 Constraint Association Construction

In section 4.4, three functional association schemes are proposed. Such functional associations are specified by a collective set of constraints, in which design knowledge can be embedded. For example, in the functional mapping scheme, when the functional associations are constructed, some basic design rule constraints will be added automatically, such as design pressure and design temperature of process equipment should be 10% higher operating pressure and operating temperature, correspondingly. Regarding specific type function and equipment, some default rules apply. In the case of pumping function association construction, if a pump is selected, the following design rule constraints will be created.

$$FR > 0.75 * BEP$$

$$FR < 1.25 * BEP$$

$$NPSHr > NPSHa$$

Where

- FR is operating flow rate;
- BEP is the best efficiency point of the pump;
- NPSHa is the available net positive suction head;
- NPSHr is the required net positive suction head.

In this scheme, engineers are still allowed to modify the constraint which are automatically added, or add more user-defined constraints.

In the special design scheme, constraints to be employed are not possible to be predefined, and hence, most constraints have to be added by engineers. Engineers are allowed more freedom to define specific constraints to quantify the relationships among engineering entities. In this case, to create a new constraint association, the user first has to select the feature that the constraint is attached to. Once it is selected, a constraint creation user interface is popped out to allow engineers to specify details, involving parameters governed by this constraint, expression and its strength index. The comments are optional, providing a descriptive annotation on why the constraint is imposed, which will benefit future reference. The feature hierarchy is available to assist engineers to quickly address the governed parameters. After the details are confirmed, the following three procedures are performed by the constraint_association_construct () method.

- A constraint node is generated with the detailed specifications stored into the central database.
- A relationship between the governed parameters and this constraint is created, i.e. the connection between the parameter nodes and the newly created constraint node is added into the edge lists.
- A validation mechanism is invoked to check the validity of current values of the governed parameters against the constraint.

6.5 Design Model Validation

Due to the inherent associations across disciplines, the engineering designs in individual disciplines are mutually constraining. Thus, a sound mechanism to represent and validate against non-geometrical relations is required to validate feature models. Compared to the validity mechanisms of B-rep or CSG, feature-based modeling is currently still weak in this aspect (Ma et al., 2009).

Once there is any operation to a feature or constraint instance, it has to be revaluated to ensure design consistency. When a new constraint is inserted or modified, a validation procedure has to be invoked to check whether the current value of the governed parameters is consistent with the constraint. It needs to be noted that, in the extensional expression constraint, the validation is performed by finding whether current values of governed variables of this constraint match with any of the specified satisfying tuples. In the case that the value of a feature parameter is changed, the associated constraints of this parameter are identified. Then, the updated value of the feature parameter has to be checked against all its associated constraints with current value of the other variables governed by these constraints.

When the conflicts happen to the preliminary mapping scheme, new equipment will be selected to meet the update function requirement. When the conflicts happen to the latter two schemes, if the current values of the feature parameters are found to be violating against the constraint, a further change propagation mechanism will be invoked to find a new consistent design solution, as shown in detail in the next chapter.

In general, two levels of evaluations have to be done to validate engineering design models of chemical process projects. The first is the intra-discipline consistency validation, which checks whether constraints imposed in individual disciplines hold; while the other is to check interdisciplinary consistency.

The engineering design is interdisciplinary consistent if:

- Each unit process feature is linked with at least a required function via a valid dependency association, and the constraints specifying relationships among parameters of functions and chemical process features hold;
- Constraints specifying inter-domain functional associations hold and each one required has its valid mapping provided function;
- Each provided function is associated with one or some equipment components or mechanical features and constraints specifying relationships between provided function and mechanical features hold.

In the proposed unified engineering framework, the low level geometrical relations, e.g., parallelism of faces or lines, distance constraints, or fixed coordinates, are not managed and solved in the same level. Rather, high-level relations, e.g. functional relations, are solved first; while the geometrical model is driven by the parameter set, which is associated with high-level mechanical features.

6.6 Dynamic Feature Parameter Association Map Generation

6.6.1 Feature Parameter Association Map

The feature parameter association map is constructed in the form of a bipartite graph to show the inter-feature dependencies. A bipartite graph consists two disjoint node sets and the edge always connects the nodes from one set to another. In this paper, the association map is constructed as AM={FP, CS, E}, where FP is the node set of feature parameters, CS is the node set of constraints, and E is the set of edges that connects one node in FP to one in CS when that feature parameter is governed by that constraint. In such a bipartite graph association map, a feature parameter node does not connect with a constraint node but only connect to a constraint node. It means that one feature parameter does not associate with any other feature parameters directly but through constraints. In this case, the set of nodes connected to one constraint node represents the group of variables governed by that specific constraint. The edge could be either directional or bidirectional to represent one-way dependency and mutually constraining relationships, respectively. A directional edge pointing to this feature parameter implies that this feature parameter depends on the value of other feature parameters which nodes point to this constraint, while the change of its value does not affect the other feature parameters governed by this constraint; in contrast, a directional edge from a feature parameter node to a constraint node means it is not governed by this constraint and not affected by the value of the other parameters governed by this constraint, but instead, its change could affect the other feature parameters through this constraint. In the latter case, bidirectional edges imply that these feature parameters are mutually constraining, i.e., a change of any of the feature parameter connected with a bidirectional edge to a constraint node would either receive change impacts or propagate change impacts to the other feature parameters governed by this constraint. It needs to be noted that the parameters that are not currently covered by any constraints are skipped in the constructed association map. The advantage of the feature parameter association map is that it provides rich semantic information with not only whether any two variables are associated but also the details how the variables are associated.

6.6.2 Dynamic Association Map Generation Mechanism

The features and constraints are all systematically managed by the implemented system, and hence, the feature parameter association map can be dynamically generated whenever needed, which shows real-time association information. The pseudo code of feature parameter map generation is shown in figure 6.2.

```
Algorithm 1: Association Map Generation
Association Map (rootnode, CN)
AM = (FP, CS, Edges)
FP: Set of associated feature parameter nodes
CS: Set of associated constraint nodes
E: Set of edges connecting the constraints and their governing feature parameters
CN: Set of all constraint parameter connection list
Begin:
temP: temporary set of parameter nodes
temC: temporary set of constraint nodes
For each C in CN
    If C's scope covers rootnode then
            CS ← C
            temC ← C
    End If
End For
```

```
Do while temC ≠ Ø
      For each C in CN
             If C ∈ temC
                      For each v in GV of C
                               If v ∉ FP & v, then
                                          FP ← v
                                          temP ← v
                                          E ← {v, C}
                               End If
                        End For
             End If
      End For
      For each v in temP
                 For each C in CN
                               If C's scope covers v then
                                            CS ← C
                                            temC ← C
                                            E ← {C, v}
                               End If
                 End For
       End For
Loop
Return AM = {FP, CS, E}
End
```

Figure 6.2 Feature parameter association map generation algorithm

Figure 6.3 shows one example of generated association map. In this illustrated case, FP = {V1, V2, ... , V18}; CS = {C1, ... , C11}; E is the set of edges connecting a node from FP to a node in C, and E = {V1 → C1, V1 → C1, ... , C11 → V15}. A filtering mechanism is also provided to skip less strong constraints, e.g. SI of constraint is less than 5.

Based on such a map, associated constraints and variables can be identified to construct a constraint satisfaction problem (CSP). This process is an important and repeated step of the proposed PECSP solving method, which is elaborated in the next section.

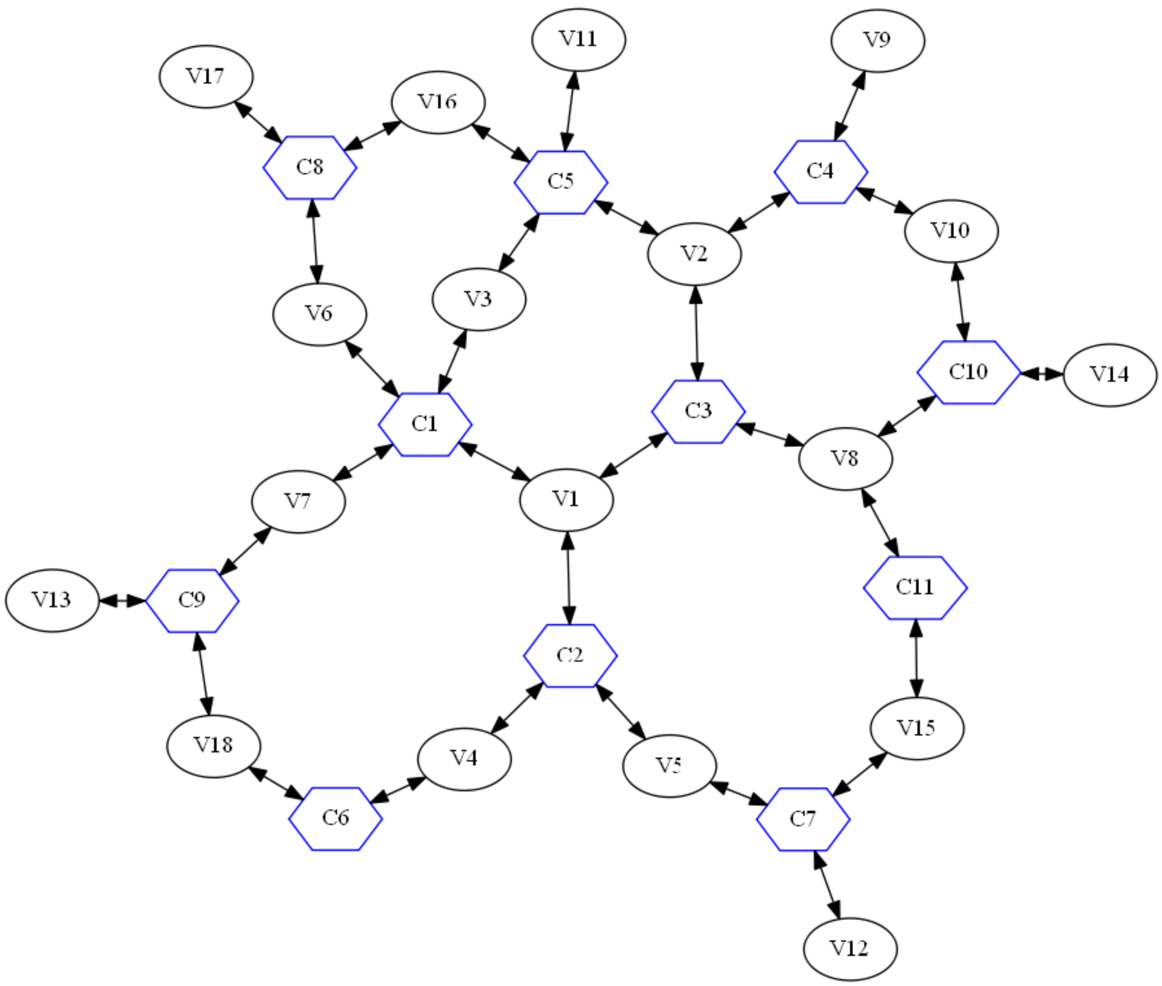


Figure 6.3 Example of feature parameter association map

6.7 Summary

In this chapter, the commonly employed variety of design knowledge in multiple domain designs, such as expert rules, design codes, numeric physical laws, are specifically abstracted into the constraint models, which express the associations among domain specific feature models in detail. The constraint models are systematically managed along with those hierarchical feature models. Consequently, a feature parameter association map can be dynamically generated with the proposed algorithm. Such an association map provides a visualization of the most up-to-date association information, which is also the basis for the proposed change propagation mechanism illustrated in the next chapter.

7 PECSP-Based Change Propagation Mechanism

7.1 Introduction

Concurrent and collaborative engineering has been a common practice to reduce time to market and to deal with the complexity with massive dependencies among engineering tasks. For example, in the chemical processing industry, engineering companies are challenged with more and more complex projects with shorter delivery time due to competition. In such engineering practice, engineering changes are inevitable and challenging because of the parallelization of engineering design activities. Many downstream designs start with an assumed design basis and/or incomplete information sets (Wiesner et al., 2011). Hence the downstream design has to be adjusted according to the engineering updates and changes initiated from the upstream. Such nonlinearity in the engineering design processes leads to frequent engineering changes. Also, iterations are involved within and across multiple engineering teams. Engineering information has to be passed down and also fed back from phase to phase, e.g. from conceptualization, to feasibility study, to initial design, and to detail design. Frequent engineering changes happen in all phases of the entire life cycle of a chemical process project. In the revamp and maintenance phases, engineering changes are even the core activities (Jarratt et al., 2011).

Generally, two types of impacts led by proposed engineering changes have to be taken into consideration. The first type is the direct impacts in a specific domain, module or discipline. This type of change impacts is relatively obvious and easy to identify and manage compared to the other one, which is the indirect impact on other engineering areas.

As is mentioned in the previous chapter, change impacts could propagate to the engineering aspects in other engineering disciplines when local validation fails. Due to the intricate associations among the engineering entities and the multi-level engineering activities in nature such as in chemical process projects, one minor engineering change could lead to undesired effects on other engineering aspects. Such a snowball effect will continue on and on until the induced engineering changes are fully absorbed such that the impacts are completely accepted by all the engineering stakeholders and will not cause any undesired effect on other engineering entities. The impacts mentioned above and possible conflicts in the design are not apparent to engineers while they have to be evaluated quantitatively in the change propagation process.

This chapter proposes a scalable engineering change propagation method, namely an expanded dynamic constraint satisfaction method to maintain the consistency of engineering models across disciplines. This method will reason on the dynamically generated feature parameter association map to find the associated variables and constraints, so that change propagation is modeled as a series of constraint satisfaction problems with progressively expanded scopes. Such iteration ends until the formulated CSP is feasible; and the solution to that CSP shows quantitative impacts on the existing design. In this procedure, the constraint solving is out of the research scope but left to commercial solvers.

7.2 Built-in Strategy of the Proposed Method

In this thesis, associations are supposed to be represented by constraints; in a project context, the number of constraints a design has to satisfy could be tremendous. There are two disadvantages to solve the whole CSP from origin, which will lead to unaffordable computation efforts. In this research, a PECSP method is proposed to solve conflicts by propagating engineering changes. The strategy of the proposed method is to minimize change impacts and computation efforts by limiting the scope of propagation. The conflicts are always attempted to be solved with formulation of localized CSPs around the engineering change source. The scope of formulated CSPs is then progressively expanded only if no solutions can be found for the previous CSP formulated. Therefore, the impact is supposed to be less than solving the global problem, since fewer modifications are made to previous design (Smith, 2005). Besides, time complexity of most algorithms for CSP solving depends on the domain size, number of variables and constraints, as well as arity of the constraints (Brailsford et al., 1999). The latter is not determined by formulation of CSP but by engineering design. However, the proposed method here tries to limit the size of the problem, which can be expected to lead to less computation efforts.

7.3 Notations of CSP

Some important notations of constraint satisfaction problem (CSP) are briefly recalled here.

A CSP is formulated as a tuple, {X, D, C} (Brailsford et al., 1999; Dechter, 2003), where

X is a set of variables {x1, x2, …, xn}. D is the domain of the problem. $D=D^{x1} \times D^{x2} \times \ldots \times D^{xn}$, where D^{xi} (i=1, … , n) is the corresponding domain for the variable xi. C is a set of constraints specifying the relations among variables of a subset of X. Note that in the proposed method, the domain of each variable is dynamically calculated based on its associated variables, i.e. by solving another CSP with look-forward variables.

Solving a CSP means either identifying that there is no feasible solution or finding a feasible solution where there exists an assignment of values to variables from their corresponding domains with no violation to any constraint (Dechter, 2003).

7.4 Local Admissible Range Calculation

Local admissible values of a feature parameter is defined here as the assignment of values from its domain without violating its directly connected constraints based on the current value of feature parameters governed by those constraints. An algorithm is proposed to calculate the range of local admissible values. This algorithm mainly serves two purposes. First, it can be quickly identified whether the engineering change leads to conflicts when it happens. Second, it serves to determine the admissible domain for the most outside variables in the formulated CSP, which is going to be further described in next section. The proposed algorithm based on the association map for range calculation firstly identifies the associated constraints and their governing variables based on the feature association map and then solved the formulated CSP to get the range of admissible values. Pseudo code of the algorithm is given in figure 7.1.

```
Algorithm 1: Local Admissible Range Calculation
Function Range (v, AM)
v: variable
AM: Association Map
Begin:
CL: constraint lists
Range: range of admissible values of v due to a constraint
TotalRange: range of admissible values of v restricted by all constraints
For each constraint c in AM,
```

If c's scope covers v And not in the scope of formulated CSP, then

Add c to CL

End If

End For

For each c in CL

Get the variables set V and their value assignment associated with constraint c

Replace the other variables except v with the assigned value & put in CL

Range = Calculate Interval (V, c)

TotalRange = TotalRange ∩ Range

End For

Return TotalRange

End

Figure 7.1 Local admissible range calculation algorithm

Once the change of a parameter happens, algorithm 1 is invoked to calculate the range of admissible values, and hence identify the existence of any design conflicts. In the map shown in figure 6.3, if the value of V1 is changed, three directly connected constraints, C1, C2, and C3, are added into the CL. Then, V3, V6, V7 are identified as the associated variables with constraint C1. The current values of {V3, V6, V7} will be used to calculate admissible range of V1 restricted by constraint C1. Similarly, {V2, V8} and {V4, V5}, are found directly associated with constraint C2 and C3, correspondingly. After the ranges led by C2 and C3 are determined, the total range is the intersection of these three ranges. The range calculation will be used to dynamical calculate the domain of variables in the formulated CSP, which will be illustrated in next section. In addition, the calculated range of the parameters can provide decision-making support by showing allowed range of the variable without any impacts to current design. The engineering design is free of affecting any other engineering designs.

7.5 Encoding Engineering Change Conflict Resolution into CSP

In each cycle of the PECSP method, a CSP is formulated to reassign values to each parameter which can satisfy all existing and possibly updated constraints. The variables and constraints are searched in the dynamically constructed feature association map to check whether they should be involved in current formulated CSP. First, it is needed to define the term “level” in the search process.

Definition 1: Level of constraints or variables specifies their distances to the source node in the search of association map. In this research, the distance of each edge in the association map is assumed to be 1. Therefore, level n of constraints or variables are the constraint or variable nodes which distance to the source node is equal to or less than 2n.

The CSP is incrementally formulated until a formulated CSP is found to be feasible. In this cyclic working loop, at cycle i, the engineering change conflict problem can be encoded as a CSP is a tuple, {FP, AD, AC}, where

FP is a finite set of feature parameters which level is equal or less than i.

AD is the admissible domain of PECSP, which is the intersection of general domain (GD) of each associated variable and the admissible range (AR) determined by a deeper level of associated constraints, i.e. $AD = GD \cap AR$. It needs to be noted that AR is only calculated for the variables with largest level, i.e. the most outside variables in the scope of formulated CSP.

AC is the finite set of associated constraints which level is equal or less than i in the association map.

7.6 Progressive Expanded CSP Solving Procedure

The procedure for the PECSP method is shown in figure 7.2. Firstly, the initial CSP is formulated. If no feasible solution can be found for the CSP of this level, i.e. conflicts cannot be solved within this level, a new CSP, which is the expansion of previous CSP with addition of one further level of variables and constraints, is reformulated. New CSPs are progressively formulated and solved until a feasible solution can be found. The expansion of the scope of formulated CSP is based on the search of the feature parameter association map. It needs to be noted that the search here is different from the search in CSP solving, such as backtracking search algorithm. The former searches in the constraint network to determine the scope of CSP, while the latter searches in the solution space.

In the process of change propagation, two different situations need to be treated separately. The first scenario is value of a feature parameter is changed; the other one is change to the constraints. As is shown in figure 7.2, the differences of handling the above two scenarios lie on the initial treatments, which are illustrated in detail, accordingly.

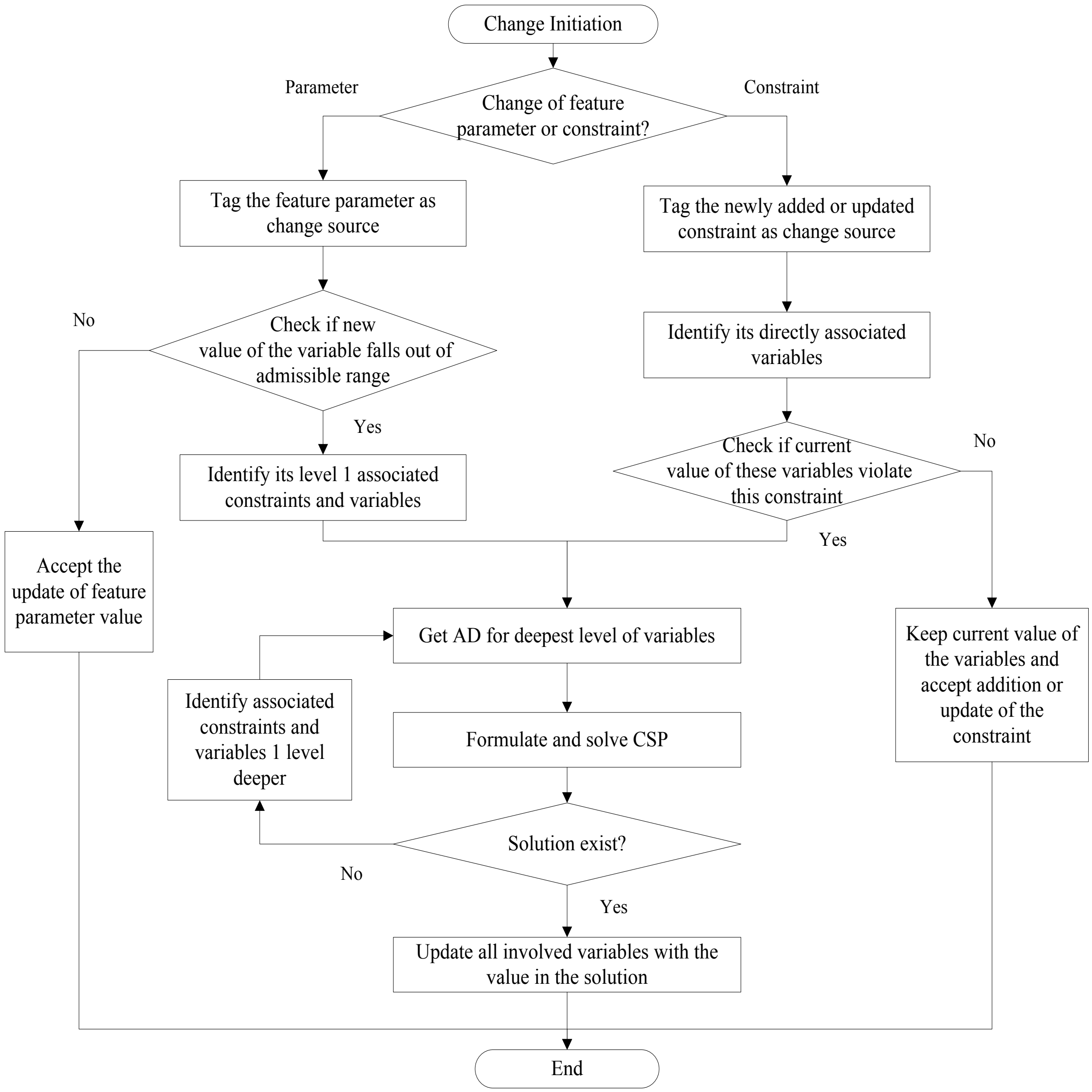


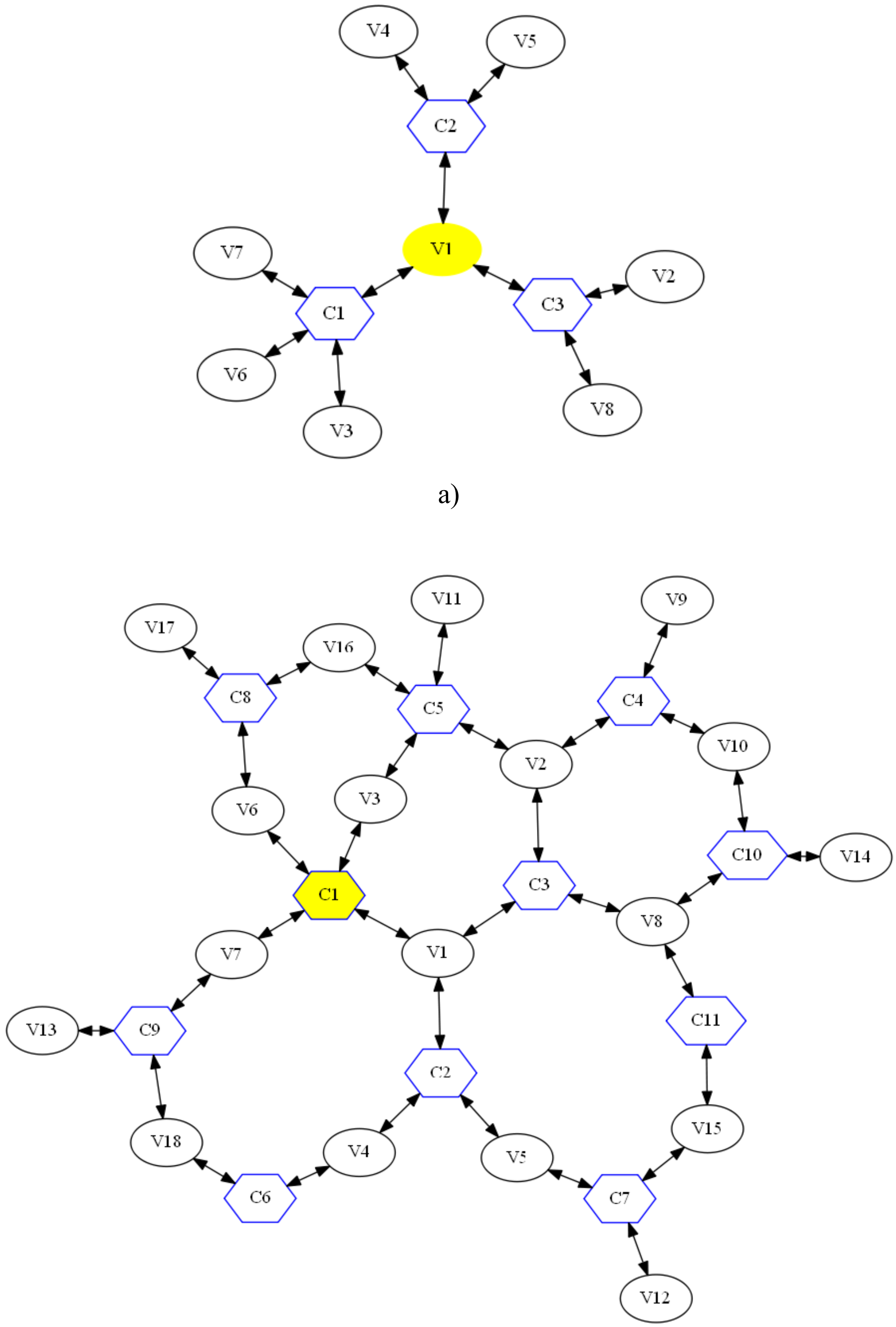
Figure 7.2 Procedures for change propagation

7.6.1 Procedures for Change to a Parameter

In such case, the new value of the parameter will be validated against the admissible range. If it falls within the range, the update to the feature parameter will be accepted; otherwise, its level-1 associated constraints and variables are then to be identified to formulate level 1 CSP. In the latter case, the system will search for a further level of associated constraints as well as their governed variables to this change source. Also, further level constraints and variables associated are identified to calculate admissible ranges for the deepest level variables in the formulated CSP and hence to determine their admissible domains. Thus, the admissible domains of the variables in the formulated CSP will be less than or equal to their original domain of those variables. If there exists a solution to this CSP, then the system can recommend this solution to engineers and update the assignment to these variables with this solution. Otherwise, a new CSP will be formulated, in which a further level of constraints and variables will be involved. This formulation process repeats until the formulated CSP has a feasible solution. In each cycle, the admissible domain is only calculated for the deepest level of variables in those formulated CSPs.

In the example shown in figure 6.3, if the change to V1 falls out of range, the search will be similar to that in the calculation of admissible range. The difference is that the associated variables found, {V2, V3, … , V8} are considered as variables in this initial formulated CSP with no assignment of values. The AC is {C1, C2, C3}. The AD to each variable in FP is determined based on the algorithm proposed in section 6.3. The scope of this iteration is shown as figure 7.3 (a).

If a feasible solution to the above CSP can be found, then the solution is given. If not, the scope of search is expanded from the scope of previous CSP. Then a new CSP is formulated and solved. The above steps are repeated until the formulated CSP has a solution. The scope of the formulated problem for the second iteration is shown as figure 7.3 (b).



a)

b)

Figure 7.3 Scopes of the formulated CSP problems when the value of V1 is changed; a) iteration 1 (8 variables & 3 constraints); b) iteration 2 (18 variables & 11 constraints)

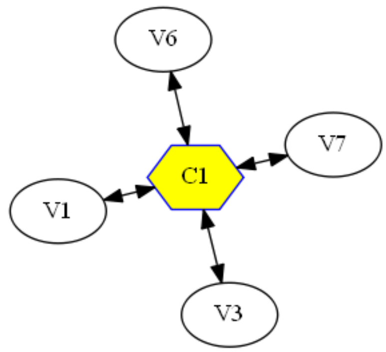
7.6.2 Procedures for Change to a Constraint

The other case is that a new constraint is added or an existing constraint is modified. In this case, search will start from the constraint node and look for the associated feature parameters. The current values of those parameters are validated against the new constraint specified. If all these associated variables are not against the constraints, the insertion or update of this constraint is accepted; otherwise, the parameters that violate the constraint are considered as variables and added into FP of the formulated CSP. The rest of steps are same as the case of change to a feature parameter.

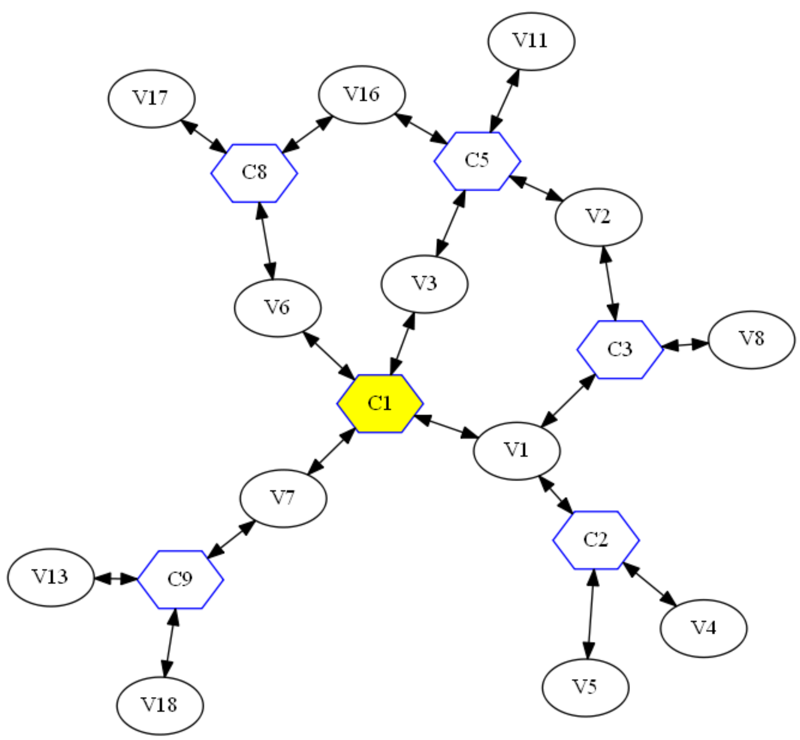
For example, if C1 in figure 6.3 is changed, current assigned values of C1's directly associated variables set {V1, V3, V6, V7} are then used to check whether they violate the new C1. If so, FP of the initial CSP is then {V1, V3, V6, V7}, and AC then equals {C1}, as is shown in figure 7.4(a).

In this illustration, the variables governed by the constraint C1 do not change. This does not hold true for all cases. For example, C1 could be changed to only govern {V1, V3} instead of {V1, V3, V6, V7}. The major advantage of the method in this thesis is that the feature association map is always dynamically generated to guarantee that it reflects most up-to-date association information. This is especially necessary whenever there is a change to the constraint in case the scope of its governed variables changes.

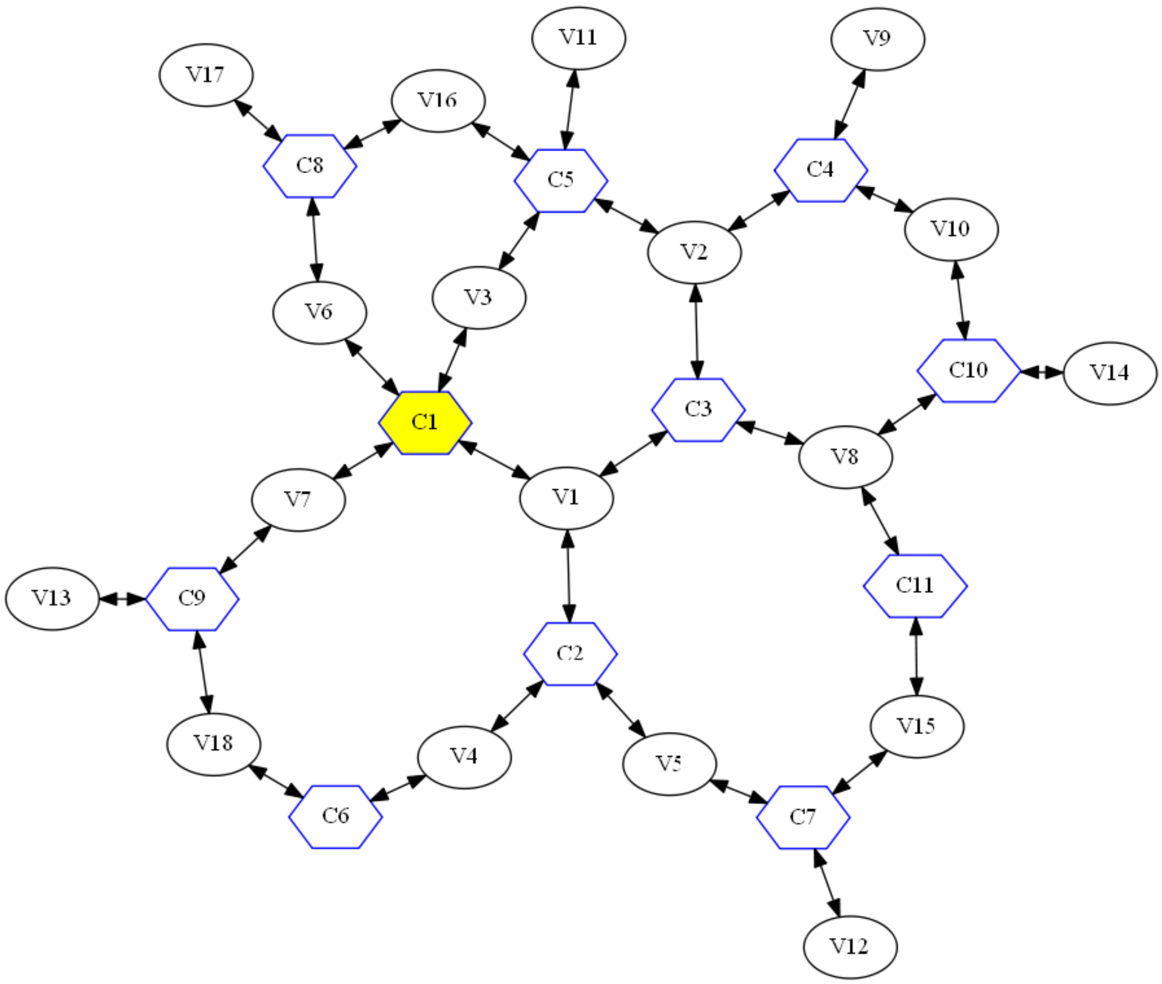
If this initial CSP has no solution, search is then expanded from the current scope. Constraints {C2, C3, C5, C8, C9} and variables {V4, V5, V8, V2, V11, V16, V17, V13, V18} are then involved, as is shown in figure 7.4(b). If the second CSP still has no solution, a new CSP with the scope of figure 7.4(c) is then formulated.


V6
V7
C1
V1
V3

a)


V11
V17
V16
C8
C5
V2
V3
V6
C3
V8
C1
V1
V7
C2
V13
C9
V4
V5
V18

b)



c)

Figure 7.4 Scopes of the formulated CSP problems when constraint C1 is changed; a) iteration 1 (4 variables & 1 constraints); b) iteration 2 (13 variables & 6 constraints); c) iteration 4 (18 variables & 11 constraints)

7.7 Summary

A PECSP solving method is proposed in this chapter, which automatically formulates the change propagation as a series of CSPs. Based on the dynamically generated feature parameter association map, the related constraint satisfaction problems (CSPs) are to be incrementally and dynamically identified, constructed and solved to implement engineering change while eliminating potential conflicts. Initially, the change source is identified as the root, and the range calculation algorithm is proposed to quickly detect design conflicts based on the identified associated parameters. If the value falls out the admissible range, engineering change propagation is invoked. Based on the identified associated nodes, the variables and their domains involved will be dynamically identified; thereafter the CSPs will be established and solved to find assignment of each variable that satisfies the constraint models.

It needs to be noted that the proposed method is not intended to replace engineers but rather to assist engineers with locally-controlled and hence efficient engineering change evaluation results or conflict resolving suggestions on engineering change implementation. This proposed method provides a well-controlled pre-processing mechanism to formulate problems for CSP solvers. There are two major advantages of this method. First, it can be expected that the impacts and computation efforts will be less than solving the global problem since the scope of the problem is always limited to the least possible. Such feature also makes this proposed method scalable to handle large industrial projects, since it always attempts to solve the CSP with the smallest scope no matter how large the project is. The variables and constraints involved in the CSPs are incrementally added until the current CSP has a solution. This can save so much computation efforts that the solution suggestions can be provided to engineers in a pretty timely fashion. Second, the solution generated tends to need fewer modifications to the existing design, which will be a great advantage to engineering change implementation, such as in the design retrofit (Smith, 2005).

8 Prototype System Implementation

A prototype system was developed with Microsoft Visual Studio 2008 to establish a design environment that supports collaborative engineering in chemical engineering projects. The prototype system has worked with several preliminary functions. It can demonstrate the feasibility of such cross-domain associations and checking in design processes proposed in this research. To show how the system works, a typical oil sands upgrading process was used for a demonstration (Chen & Munteanu, 2012; Gray 1994). The simplified P&ID model of this process developed with Smartplant P&ID is shown in figure 8.1. The feed, usually diluted bitumen, is preheated and pumped into the diluent recovery unit to remove diluents from the feed. Then the bitumen goes to the vacuum distillation unit and is separated into different fractions. The lightest fraction, naphtha, is preheated and fed into the naphtha hydrotreating unit. Similarly, the diesel and gas oil fractions are fed into corresponding units to be hydrotreated. The residual is heated to its thermal cracking temperature and fed to a delayed coking unit to break the long-chain molecules, and then the lighter composition goes to the corresponding hydrotreating unit, accordingly.

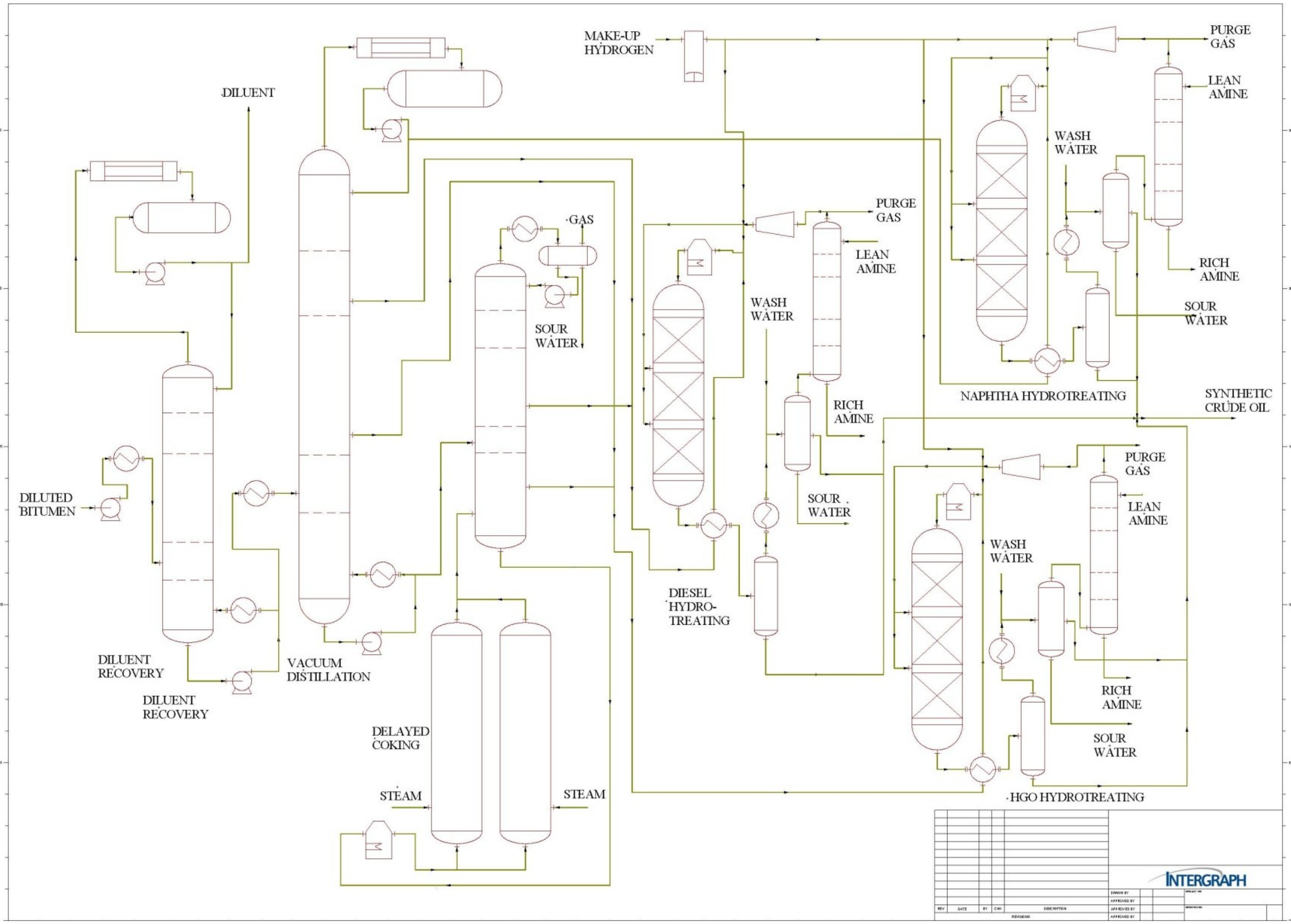


Figure 8.1 A simplified P&ID of bitumen upgrading

8.1 Feature-Based Design with the Support of the Prototype System

Figure 8.2 shows the architecture of the feature-based collaborative design prototype system. The design semantics are well hierarchically structured and can be accessed by domain engineers. As is highlighted with red rounded squares and connecting arrows in figure 8.2, the engineering entities in the chemical process engineering and mechanical engineering domains are all associated. Such easy access to the inter-domain information will significantly reduce the communication hassle while providing domain engineers precise contextual semantic information.

The system consists of three modules, namely "chemical process features management module," "inter-domain functional features management module," and "mechanical design features management module." Currently, the system offers the functionality of feature-based design and manages associations among feature models.

The chemical process features management module offers chemical engineers interfaces to add new feature objects interactively. As shown in figure 8.2, (a) and (c) add a new unit chemical process feature and a modular chemical process feature, respectively. These functions are implemented by the method embedded in the feature class defined in chapter 3. For example, the input from (a) is implemented by the function of "add" embedded in the method of the "unit chemical process feature" class and the added feature appears as a node in (b). It is noted that the modular process feature can also be added under another modular chemical process node. To do this, the user just needs to select the node and then click the "Add/Add Modular Chemical Process Feature" to initiate the new feature-adding user interface (UI). The system supports several methods of information input, such as importing data from exported files of some commercial software packages. It may also require some input from engineers, similar to what is required for documentation or input in other software packages. For example, such feature models can also be extracted from the P&ID model shown in figure 8.1. The relationships among these chemical process features are managed in the background and stored automatically. Such hierarchical structure information is systematically managed, as is shown in figure 8.2 (b).

After the chemical process is specified, the "Function_Add" UI can be initiated to add corresponding functions. Similarly, this function is implemented by the method embedded in the "inter-domain functional feature" class. The function added is associated with the chemical process automatically, as is shown by the link between (b) and (d) in figure 8.2. Then, the functional association can be built between the chemical process and the mechanical system by initiating the "Functional_Association" UI. The required functions added before are displayed in the left checklist box, while the mapping equipment is listed in the right checklist box, as is shown in Figure 8.2 (e). For example, all the matching pumps supporting the pumping function are listed in the right checklist box. The association will be built by selecting the equipment and clicking the "Build Association" button. As is shown in figure 8.2, the pump selected in (e) appears in the mechanical system (f). Such association information is stored in the database for change propagation reference. In some cases, off-the-shelf equipment may not be available for a specific function, e.g., hydrotreating, so the users can also select "Start a New Design." In this case, the association is built automatically between the required function and the new design model. It is noted that the system allows multiple functions to be checked, i.e., multiple functions will be mapped to one piece of equipment. This is to support the trend of process intensification, which may combine multiple operations into a single piece of equipment (Stephanopoulos & Reklaitis, 2011). In some design change cases, the previously selected equipment cannot satisfy the required functions. For example, some parameters of chemical process are increased, or the engineers may think alternative equipment would be a better choice based on other considerations. Then, the users can also remove the associations. This function provides engineers flexibility in handling the associations across domains; while the change will be validated to the association constraints by the system. With systematic management of inter-domain dependencies, engineers can have peace of mind because all the pieces of dependencies are tracked, managed, and evaluated systematically. Hence, the information overload can be resolved instead of becoming a new issue.

This system prototype offers similar functions for the mechanical design domain and also organizes the information into a hierarchical structure, as shown at the bottom of figure 8.2 (f). The Mechanical Design View ((g) and (f)) provides an interface for designers to further refine the pre-generated design model. The feature-based prototype system allows engineers to work on different granularities, e.g., either on the reactor level or its subcomponent level, while feature models from different levels are also vertically associated. There are also mechanisms embedded in mechanical design features to check whether they satisfy the constraints imposed in the process conceptual design phase. For example, the volume requirement of the pressure vessel specified, e.g., larger than 100 liters, needs to be satisfied by a list of attributes of mechanical design features, such as the dimensions of the pressure vessel, or the thickness of the pressure vessel must satisfy the ASME code. These "mechanical design features" are associated with CAD solid models.

For example, the pressure vessel feature is associated with the 3D model built with Siemens NX 6, as is shown in figure 8.3. It needs to be noted that figure 8.3 only shows part of the Excel file. The mechanical engineer can update the attributes based on engineering calculations through the interfaces provided by the system. The consistency will be checked, and the changes can also be reflected in the CAD models, which will be shown in the next subsection.

The associations are systematically managed by the prototype system. With predefined constraints built into the system, the system can intelligently assist engineers to find compliance off-the-shelf equipment, i.e. pumps in the case, as is shown in figure 8.2(e). In the case of no matching equipment available, the system can then lead the engineers to start a new design. In this scenario, the engineers can insert customized constraints following a specific format to establish the inter-domain association. Whenever a design change happens, the inconsistencies can be identified with the constraints built into the feature models, and changes can be propagated along this path, which is further illustrated in the next section.

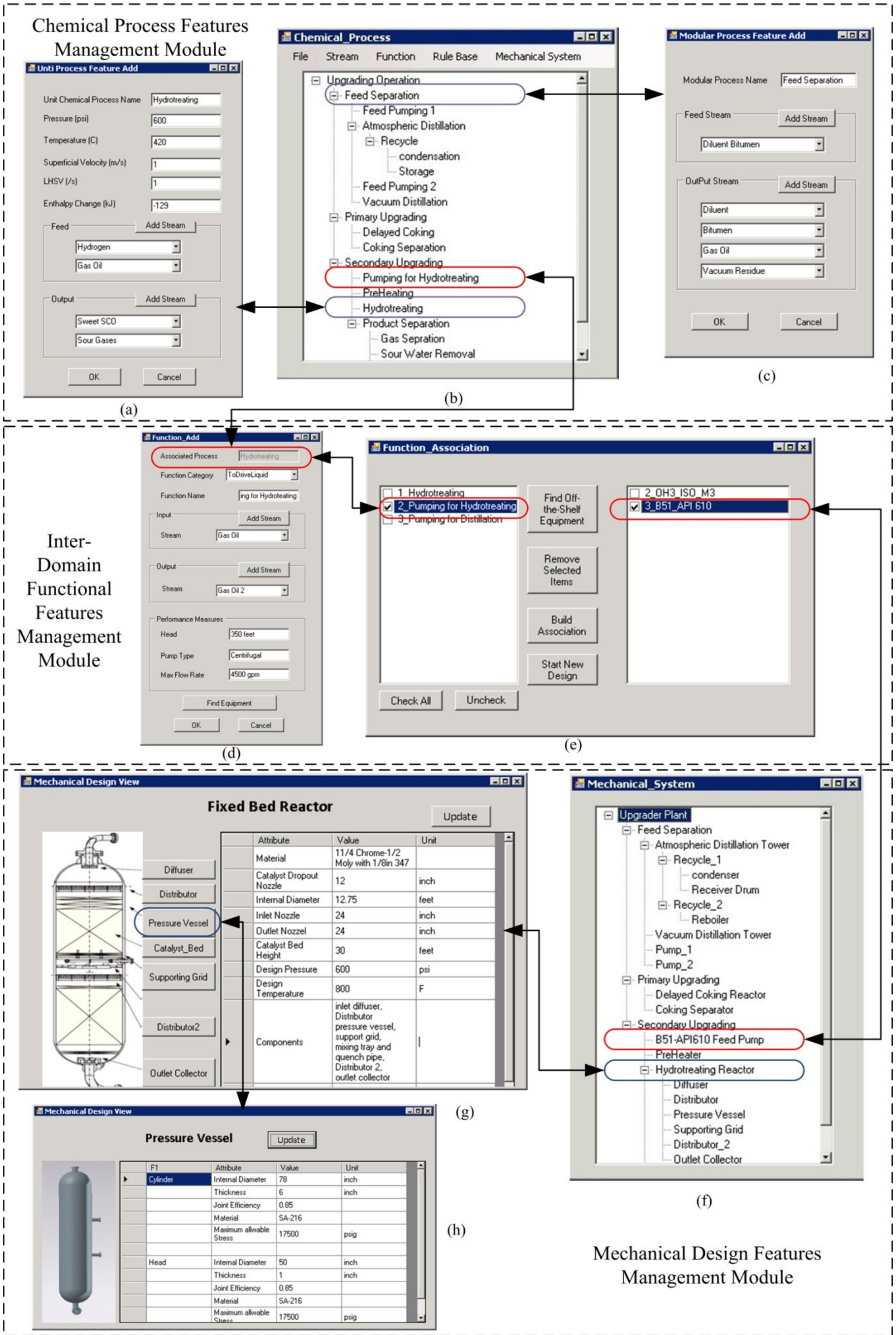
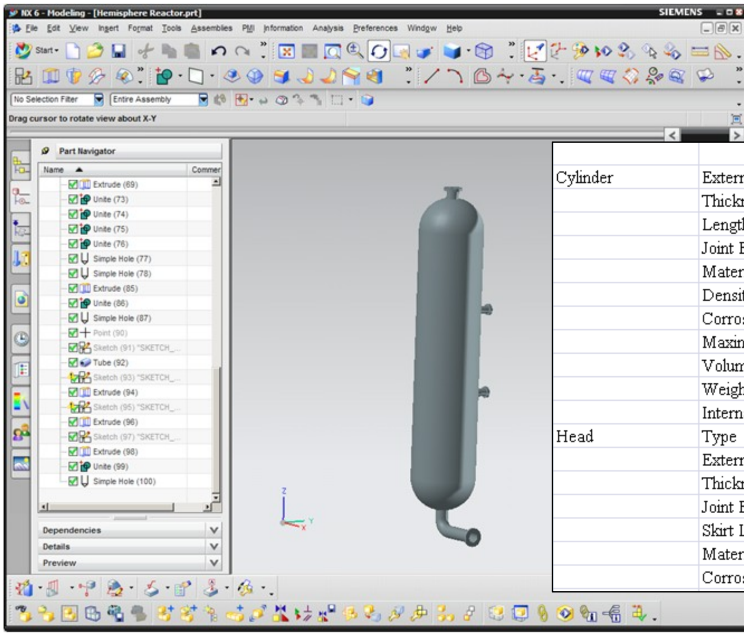


Figure 8.2 Feature-based collaborative design prototype system

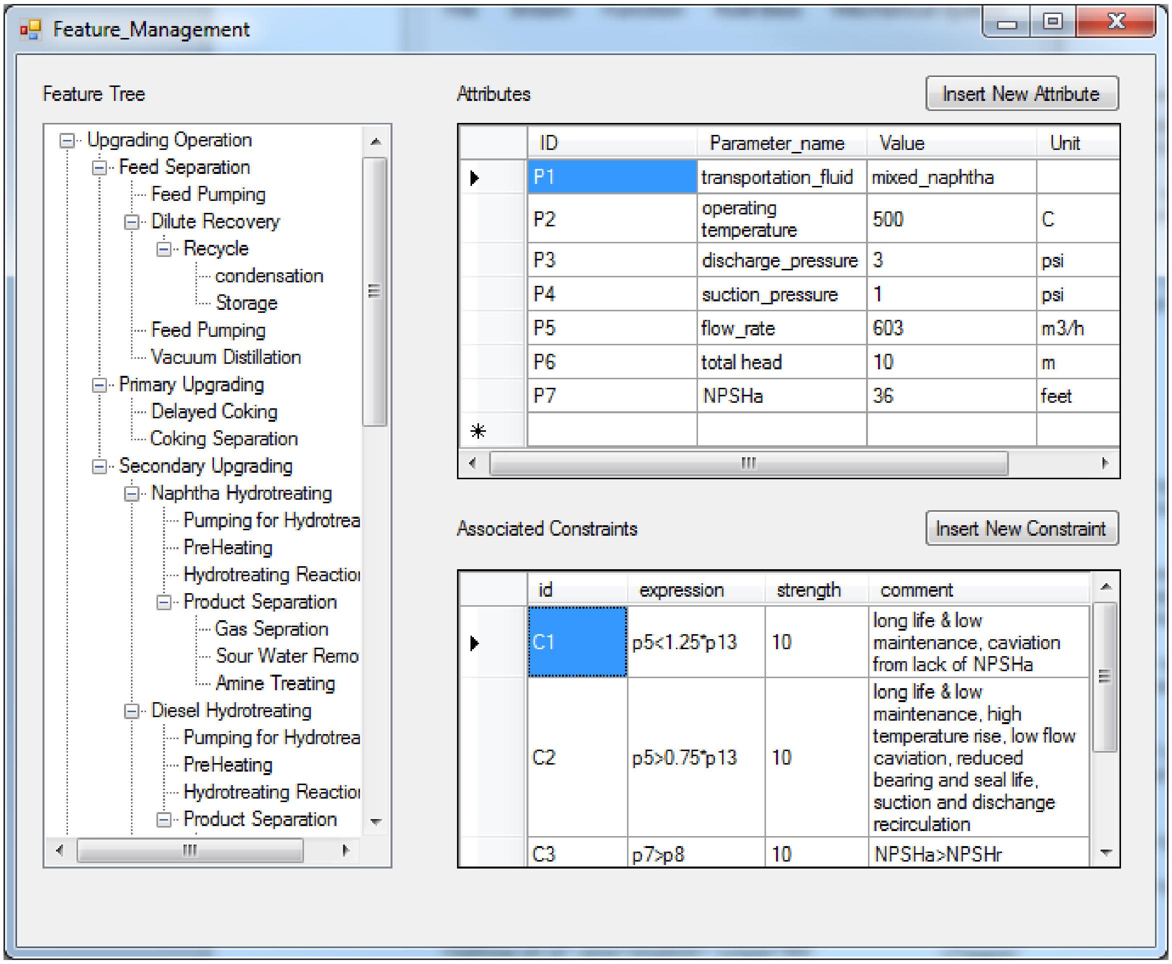


	Attribute	Value	Unit
Cylinder	External Diameter	26	feet
	Thickness	0.5	feet
	Length	100	feet
	Joint Efficiency	0.85	
	Material	SA-204	
	Density	40.84	
	Corrosion Allowance	0.1	feet
	Maximum allwable Stress	17500	psig
	Volume	28.40705163	cuft
	Weight	1136.012994	lb
	Internal Diameter	25	feet
Head	Type	Elliptical	
	External Diameter	26	feet
	Thickness	0.5	feet
	Joint Efficiency	0.85	
	Skirt Length	2	feet
	Material	SA-216	
	Corrosion Allowance	0.1	feet

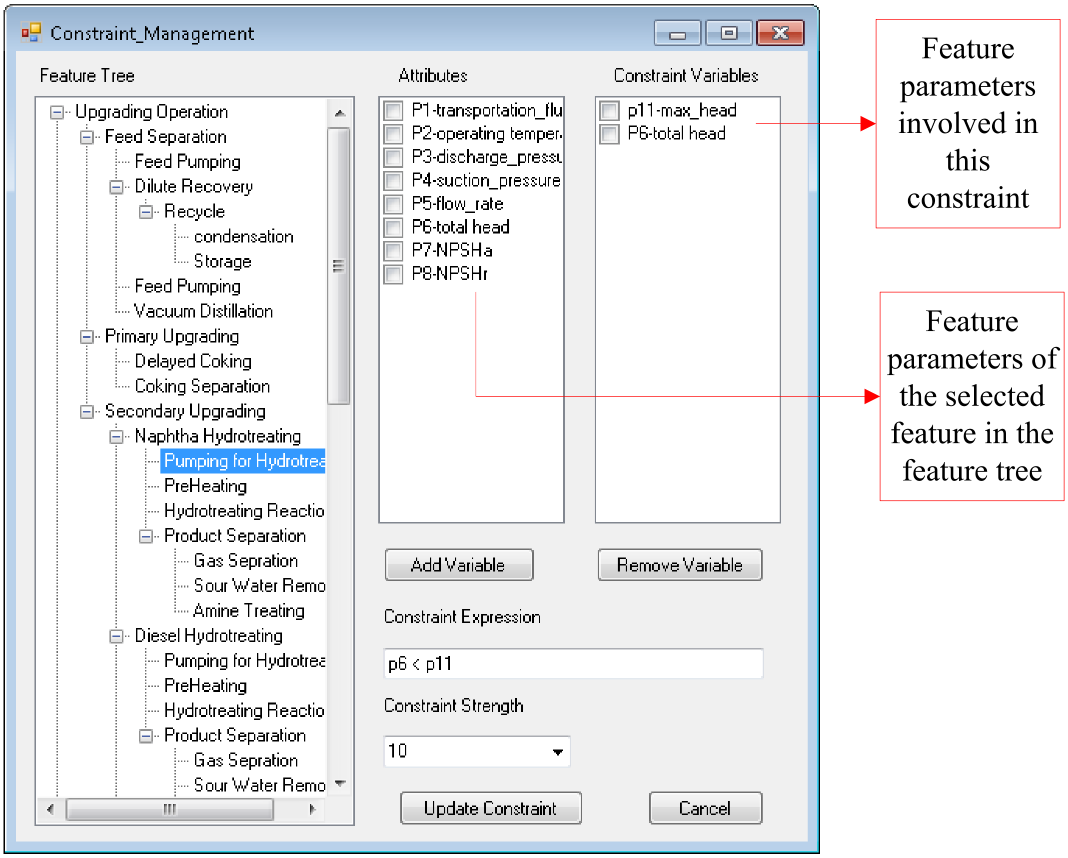
Figure 8.3 Parametric CAD model of pressure vessel for hydrotreating reactor

8.2 Systematic Constraint Management

In the implemented prototype system, a function of feature management is provided to allow engineers to dynamically add, edit, or delete feature parameters and constraints of the features. As is shown on the left hand side of figure 8.4, the features are managed in a hierarchical tree structure, each of which contains a set of parameters listed in the "attributes" box along with their IDs. The feature tree can facilitate engineers quickly finding specific parameters. Engineers can select the parameters of specific features and explicitly specify the associations among these parameters with their IDs.


Feature_Management
Feature Tree
Upgrading Operation
Feed Separation
Feed Pumping
Dilute Recovery
Recycle
condensation
Storage
Feed Pumping
Vacuum Distillation
Primary Upgrading
Delayed Coking
Coking Separation
Secondary Upgrading
Naphtha Hydrotreating
Pumping for Hydrotrea
PreHeating
Hydrotreating Reactioi
Product Separation
Gas Sepration
Sour Water Remo
Amine Treating
Diesel Hydrotreating
Pumping for Hydrotrea
PreHeating
Hydrotreating Reactioi
Product Separation
Attributes
Insert New Attribute
ID | Parameter_name | Value | Unit
P1 | transportation_fluid | mixed_naphtha |
P2 | operating temperature | 500 | C
P3 | discharge_pressure | 3 | psi
P4 | suction_pressure | 1 | psi
P5 | flow_rate | 603 | m3/h
P6 | total head | 10 | m
P7 | NPSHa | 36 | feet
Associated Constraints
Insert New Constraint
id | expression | strength | comment
C1 | p5<1.25*p13 | 10 | long life & low maintenance, caviation from lack of NPSHa
C2 | p5>0.75*p13 | 10 | long life & low maintenance, high temperature rise, low flow caviation, reduced bearing and seal life, suction and dischange recirculation
C3 | p7>p8 | 10 | NPSHa>NPSHr

a)



b)

Figure 8.4 Constraint management interfaces (a & b)

The prototype system systematically manages all the features as well as the constraints, which specify the associations among features. In the case of hydrotreating process as shown in figure 8.5, Part of the feature parameters involved are listed in table 8.1; while the constraints which that govern the associations among those parameters (ASME, 2004; Towler, 2013) are illustrated in table 8.2. The association map of the change parameter of "flow rate" can be automatically identified by the prototype system, which will further output graph information to Graphviz^{TM} to generate a graphical view of the map, as shown in figure 8.6. Limited by the size, only a partial feature parameter association map is shown here, which reflects the constraint association within five levels in the hydrotreating process in the oilsands upgrader. In figure 8.6, the red oval shows the root of change, the white hexagons with blue line represent associated constraints, and the yellow ovals are associated variables. Such association map not only shows associations between variables, but also which constraint governs such association among these variables. Since all the association information evolves along the lifecycle of the project, such association map always shows most up-to-date information.

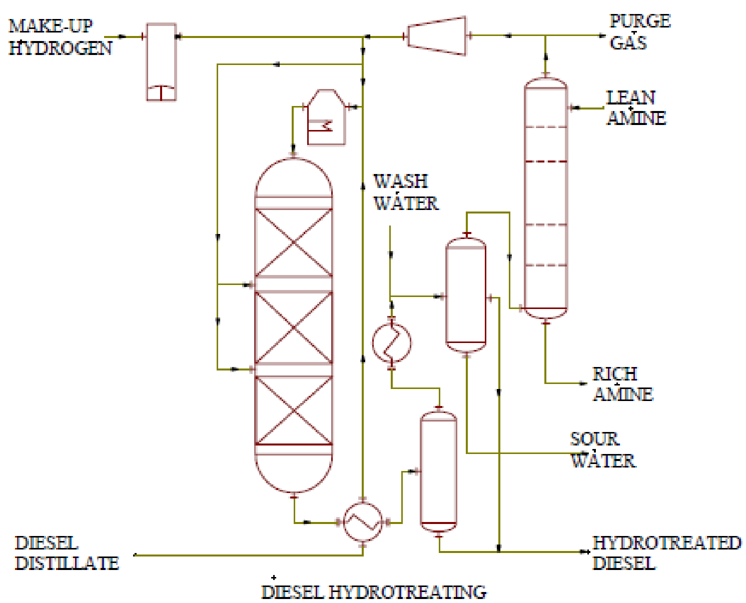


Figure 8.5 A simplified P&ID of diesel hydrotreating process

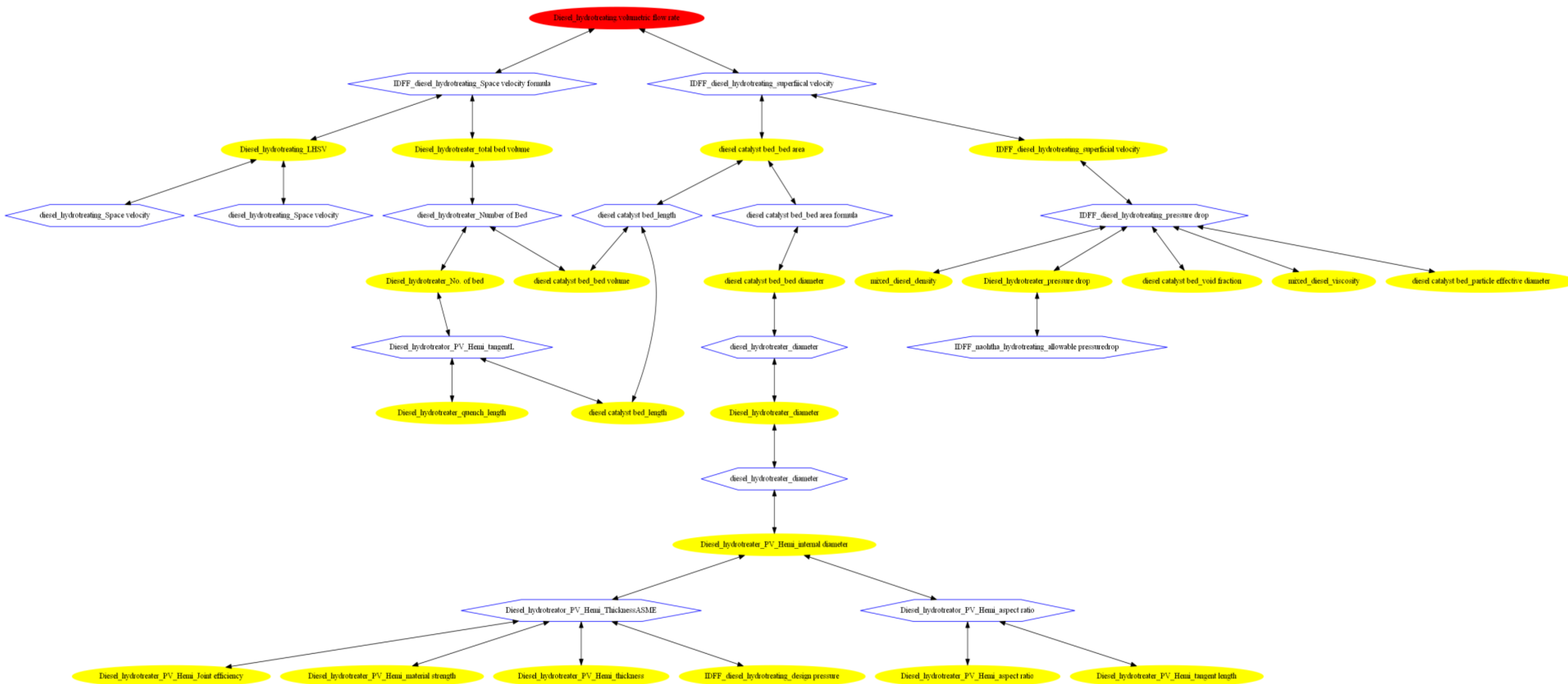


Figure 8.6 Partial feature parameter association map generated by Graphviz™ software

Table 8.1 Variables involved in the engineering design of hydrotreating process

Feature	Variable	ID	Type	Parameter Name
Hydrotreating reaction	F_v	P68	Continuous	Diesel_hydrotreating.volumetric flow rate
	LHSV	P63	Continuous	Diesel_hydrotreating_LHSV
	u	P73	Continuous	IDFF_diesel_hydrotreating_ superficial velocity
	ρ	P78	Continuous	Mixed_diesel_density
	μ	P33	Continuous	Mixed_diesel_viscosity
Hydrotreater	DP	P71	Continuous	IDFF_diesel_hydrotreating_ design pressure
	Ɛ	P108	Continuous	Diesel catalyst bed_void fraction
	A	P105	Continuous	Diesel catalyst bed_bed area
	t	P111	Discrete	Diesel_hydrotreater_PV_Hemi_thickness
	D	P81	Continuous	Diesel_hydrotreater_diameter
	SL	P106	Continuous	diesel catalyst bed_length
	TL	P123	Continuous	Diesel_hydrotreater_PV_Hemi_tangent length
	BV	P103	Continuous	Diesel catalyst bed_bed volume
	V	P96	Continuous	Diesel_hydrotreater_total bed volume
	AR	P125	Continuous	Diesel_hydrotreater_PV_Hemi_aspect ratio
	ΔP	P98	Continuous	Diesel_hydrotreater_pressure drop
	N	P95	Discrete	Diesel_hydrotreater_No. of bed
	L_q	P126	Continuous	Diesel_hydrotreater_quench_length
	S	P116	Continuous	Diesel_hydrotreater_PV_Hemi_ material strength
	E	P112	Discrete	Diesel_hydrotreater_PV_Hemi_ Joint efficiency
	d_p	P107	Continuous	Diesel catalyst bed_particle effective diameter

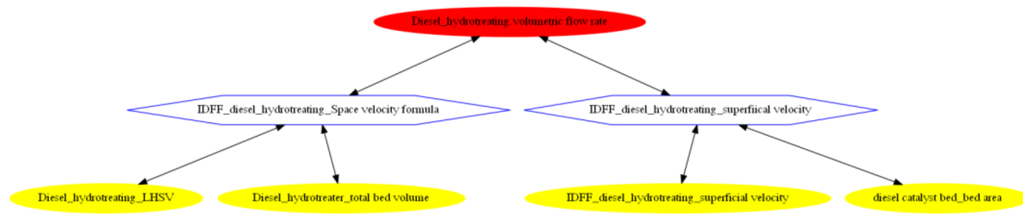
Table 8.2 Constraints associating reactor design with hydrotreating process design

Mechanistic Models	Constraint Name
$LHSV = \frac{F_v}{V}$	IDFF_diesel_hydrotreating_ Space velocity formula
$u = \frac{F_v}{A}$	IDFF_diesel_hydrotreating_ superfical velocity
Data-Driven Models	
$\frac{\Delta P}{TL} = 150 * \frac{(1-\varepsilon)^2}{\varepsilon^3} * \frac{\mu * u}{{d_p}^2} + 1.75 * \frac{(1-\varepsilon)}{\varepsilon^3} * \frac{\rho u^2}{d_p}$	IDFF_diesel_hydrotreating_ pressure drop
$LHSV > 1$	Diesel_hydrotreating_ Space velocity
$LHSV < 3$	Diesel_hydrotreating_ Space velocity
Geometry Relationships	
$AR = \frac{TL}{D}$	Diesel_hydrotreater_PV_ Hemi_aspect ratio
$TL = N * SL + (N-1) * L_q$	Diesel_hydrotreater_PV_ Hemi_tangentL
$V = N * BV$	Diesel_hydrotreater_ Number of Bed
$BV = A * SL$	Desel catalyst bed_volume formula
$A = \frac{\pi * d^2}{4}$	Diesel catalyst bed_bed area formula
Design Codes & Rules of Thumb	
$t = \frac{DP * D}{2 * S * E - 1.2 * DP}$	Diesel_hydrotreator_PV_ Hemi_ThicknessASME
$\Delta P < 100000$	IDFF_naphtha_hydrotreating_ allowable pressuredrop

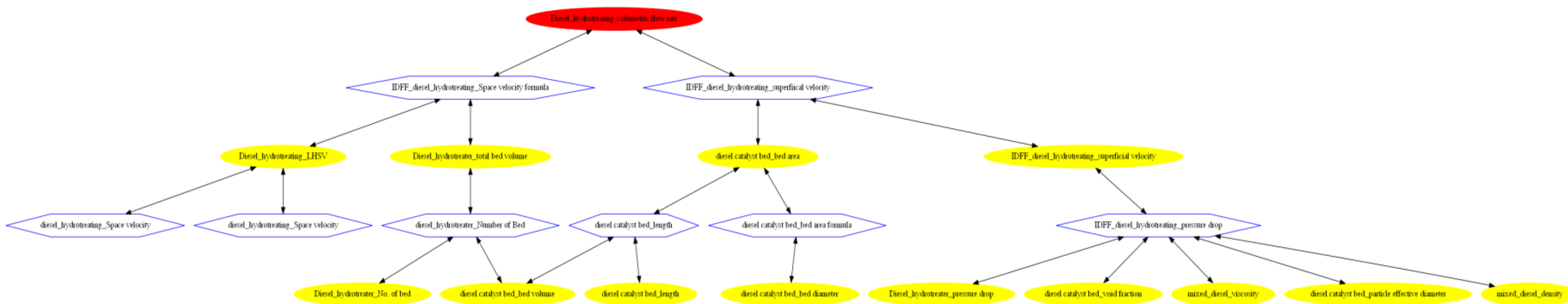
8.3 Change Propagation

8.3.1 Progressive Expanded CSP Solving Implementation

In the case of figure 8.6 that the 'flow rate' of hydrotreating process feature is changed from 602 m^3 to 900 m^3, two constraints, 'space velocity formula' and 'superficial velocity formula', are identified to associate "flow rate" with another two feature parameters, respectively. First, based on the current value of those four feature parameters and the two constraints, the admissible range of the 'flow rate' is calculated. It is found that the change of "flow rate" falls out of range of admissible values and then the PECSP method is invoked. A CSP (see figure 8.7(a)), which involves the above five variables and two constraints, is then formulated, in which the admissible domains of those five variables except "flow rate" are dynamically calculated based on their associated constraints and current value of one further level associated parameters. For example, the domain of superficial velocity is calculated based on the pressure drop formula with the current value of five other feature parameters, i.e. 'density', 'pressure drop', 'void fraction', 'viscosity' and 'catalyst particle effective diameter'. Also note that LHSV is governed by two unary constraints, dimensional constraint in this case. This CSP is solved and it is deemed that no solution exists. A new CSP is further formulated to involve one more level of constraints and feature parameters, as is shown in figure 8.7(b). The admissible domains of deepest level variables are calculated. In this case, the second CSP formulated is solvable and the solution of that CSP, i.e. assignment of values to the variables, is updated to those affected features, as shown in table 8.3. The rest of the design remains the same.



a)



b)

Figure 8.7 Scopes of the formulated CSP problems when feature parameter volumetric flow rate; a) iteration 1 (5 variables & 3 constraints); b) iteration 2 (14 variables & 8 constraints).

Table 8.3 Recomended update from the solution of the third CSP

Feature	Variable	ID	Original design	Update value	Unit
Hydrotreating reaction	Fv	P68	603	900	m^3
	LHSV	P63	1.5	2.24	h^{-1}
	u	P73	0.0059	0.0088	m/s
Hydrotreater	ΔP	P98	45801	68077	N/m^2

The recommended solution provided by the PECSP solving method satisfies all the specified constraints again by only propagating changes to three other feature parameters while keeping the rest the same as the earlier design. Meanwhile, the maximum scope of the CSP that needs to be solved only involves 14 variables and 8 constraints., which is much less than the case all associated constraints and feature parameters are solved together. Still, the performance of this proposed method needs to be further verified by real industrial case.

In this case, there is only a change to the performance parameter, "pressure drop," of the hydrotreater; while there are no changes to its physical parameters or its components. In case there is a change to, e.g. thickness of pressure vessel, after those high level features are all updated, the driving parameters, i.e. the attributes of these engineering intent features will then drive the change propagation in the low level geometrical features. The parametric solid CAD models can be updated with the support of current CAD software, e.g. NX, interfaced with our implemented system. The details will be illustrated in next section.

8.3.2 Parametric Solid CAD Model Update

In some other cases, the change may propagate to some feature parameters which are associated with solid CAD models. In the case of figure 8.7, when the operating pressure of the process supported by the pressure vessel, i.e., the maximum operating pressure for the hydrotreating reaction, is changed, the feature validation mechanism is invoked. Since new value of this parameter falls out of its local admissible range, the change propagation method is invoked to work out a new consist design solution. The variables and constraints in figure 8.7 is shown in table 8.4 and 8.5, respectively. A recommended solution is then provided as shown in table 8.6.

Table 8.4 Variables involved in the association map of figure 8.7

Feature	Variable	ID	Type	Parameter Name
Hydrotreating reaction	MOP	P70	Continuous	Diesel_hydrotreating. max pressure
	MOT	P69	Continuous	Diesel_hydrotreating _ max temperature
IDFF_diesel_ hydrotreating	DP	P71	Continuous	IDFF_diesel_hydrotreating_ design pressure
	DT	P72	Continuous	IDFF_diesel_hydrotreating_ design temperature
Diesel_hydro-treater_PV_ Hemi	tc	P111	Discrete	Diesel_hydrotreater_PV_Hemi_cylinder thickness
	E	P112	Discrete	Diesel_hydrotreater_PV_Hemi_joint efficiency
	D	P113	Continuous	Diesel_hydrotreater_PV_Hemi_diameter
	th	P114	Continuous	Diesel_hydrotreater_PV_Hemi_head thickness
	S	P116	Continuous	Diesel_hydrotreater_PV_Hemi_material strength
	$MAWP_c$	P121	Continuous	Diesel_hydrotreater_PV_Hemi _ MAWP_cylinder
	$MAWP_h$	P128	Continuous	Diesel_hydrotreater_PV_Hemi MAWP_head
	MAWP	P129	Continuous	Diesel_hydrotreater_PV_Hemi_MAWP

cor	P130	Continuous	Diesel_hydrotreater_PV_Hemi_corrosion allowance

Table 8.5 Constraints involved in the association map of figure 8.7

Constraint Expression	**Constraint Name**
$DP = 1.1 * MOP$	IDFF_diesel_hydrotreating_ design pressure
$MAWP > DP$	IDFF_diesel_hydrotreating_ MAWP
$DT = MOT + 30$	IDFF_diesel_hydrotreating_ design temperature
$MAWP = \min\,(MAWP_c, MAWP_h)$	Diesel_hydrotreater_PV_ Hemi_minMAWP
$\begin{cases} if(-20 < DT < 650)\ then\ S = 18700 \\ if(650 < DT < 700)\ then\ S = 17700 \\ if(700 < DT < 800)\ then\ S = 12600 \\ if(800 < DT < 900)\ then\ S = 6500 \\ if(900 < DT < 1000)\ then\ S = 2500 \end{cases}$	Diesel_hydrotreater_PV_ Hemi_minMAWP
$tc = \frac{DP * D}{2 * S * E - 1.2 * DP}$	Diesel_hydrotreator_PV_ Hemi_ThicknessASMEcylinder
$th = \frac{DP * D}{2 * S * E - 0.4 * DP}$	Diesel_hydrotreator_PV_ Hemi_ThicknessASMEhead

Table 8.6 Recommended update for the change to operating pressure

Feature	Variable	ID	Original design	Update value	Unit
Hydrotreating reaction	OP	P60	600	1000	psi
Hydrotreater	DP	P71	665	1104	psi
	t	P111	6	10	inch

Once these high level mechanical features are updated, it will further automatically drive the update to the parametric solid CAD models. As is introduced in section 8.1, the built CAD models are parametrically in reference to the high-level mechanical function features, i.e., the parameters of the involved geometrical features in CAD models have to refer to the parameters of high-level mechanical features. Taking the parametrical CAD model shown in figure 8.3 as an example, the partial vertical mapping relationships between high-level mechanical features and geometrical features of CAD model are illustrated in figure 8.9.

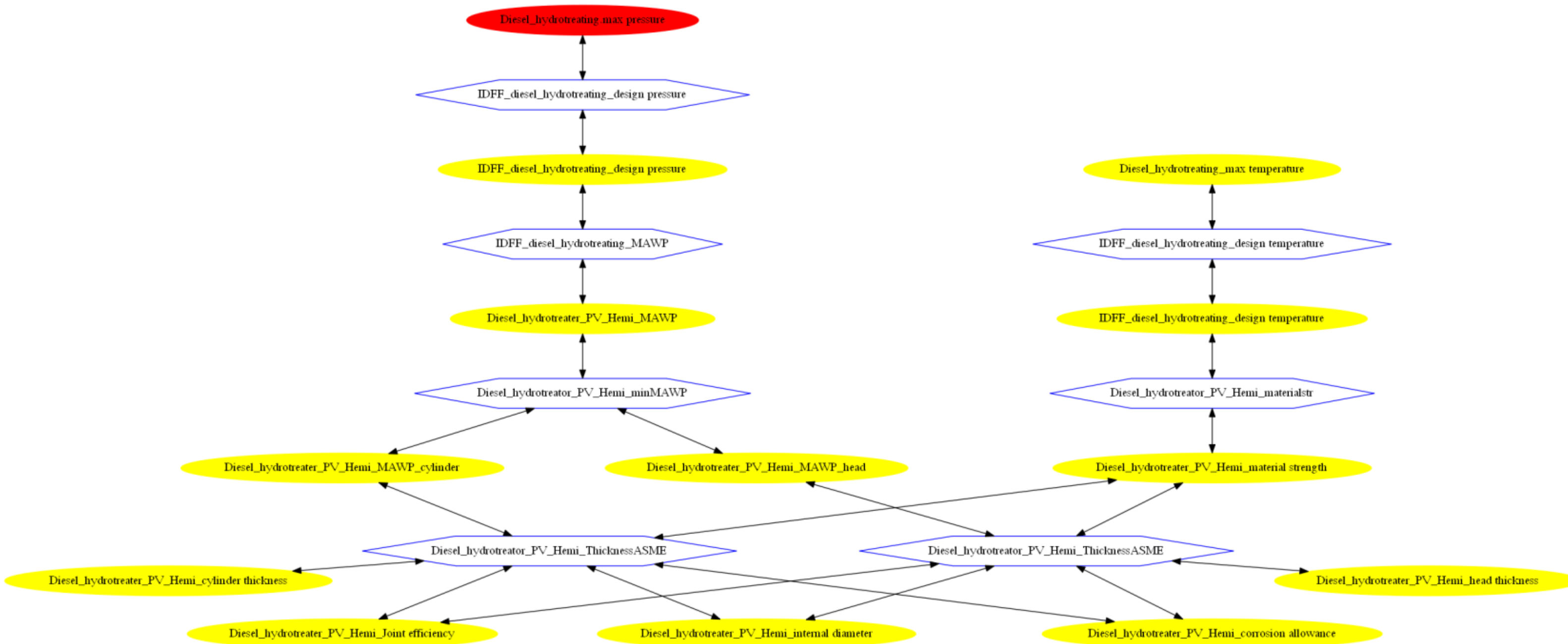


Figure 8.8 Feature parameter association map for feature parameter of maximum pressure

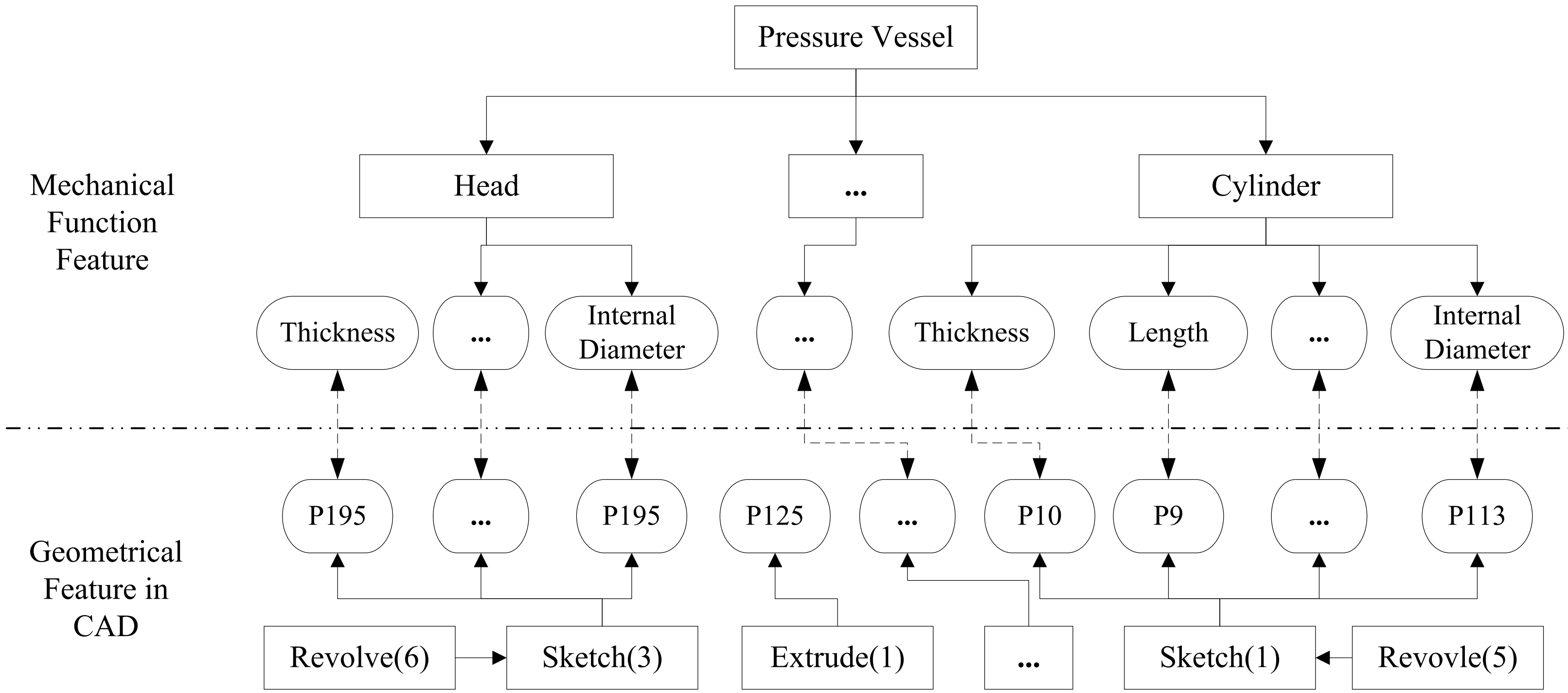


Figure 8.9 Vertical mapping between high level mechanical feature and geometrical features in CAD models

Once the changes mentioned above are accepted or updated to another value, the CAD model can then be updated by the function of "Update from External Source" offered by NX since this model is built in a parametrical way. Here the external source is the Excel spreadsheet with all the parametric information stored inside, which is managed and validated by the system to maintain consistency across disciplines with the proposed change propagation algorithm. The change to such parametric information of feature models will drive the automatic update of the CAD solid model. Part of the model after update is enlarged and dimensioned for clarity, shown at the bottom right of figure 8.10. Compared to the original design, it can be found that the thickness has changed in the CAD model of the pressure vessel, led by the chemical process design change. Such change propagation capability will keep the data consistent and save a lot of effort when design change is needed.

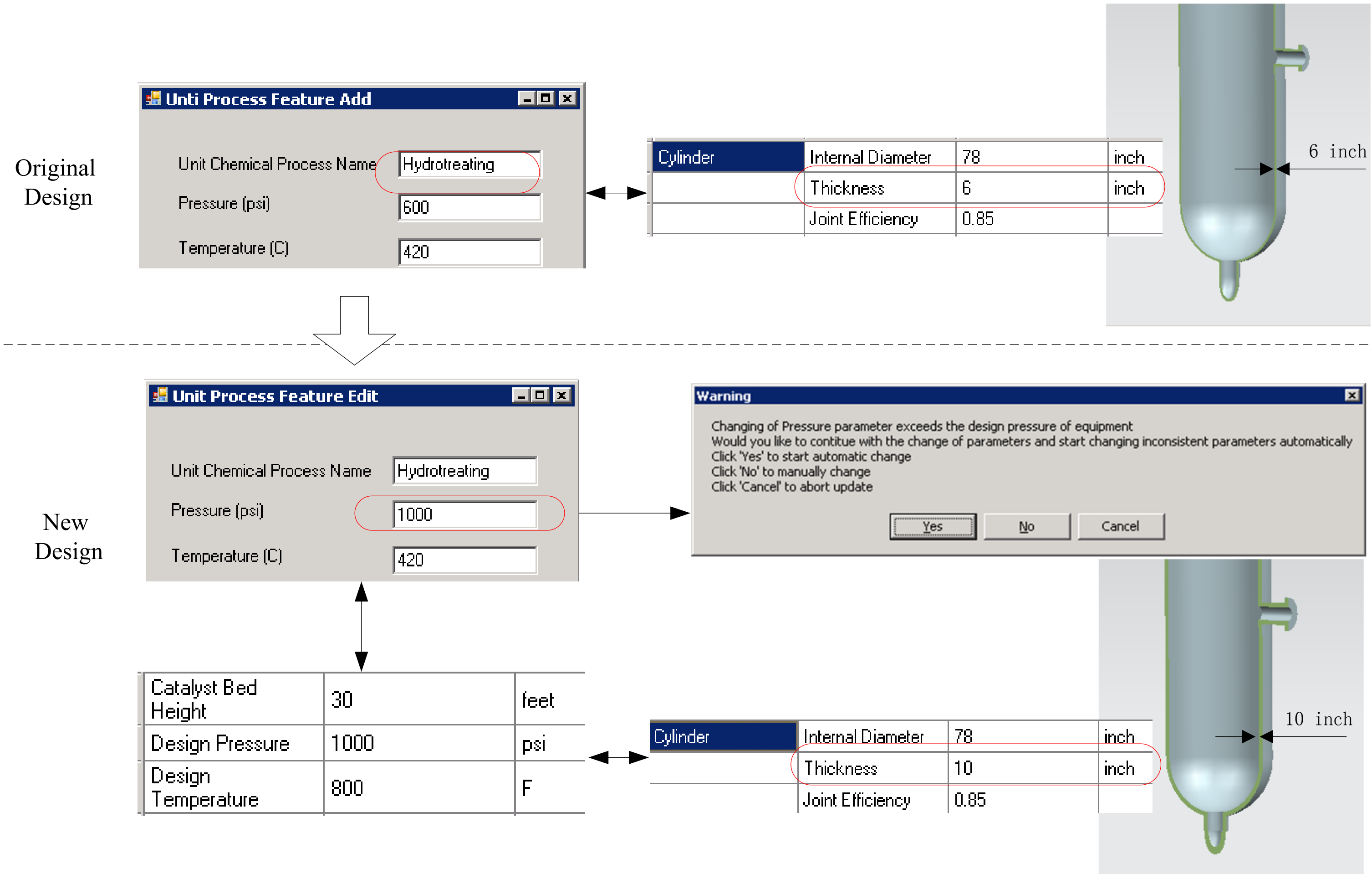


Figure 8.10 Illustration of parametric solid CAD model update

8.4 Summary

The proposed system has highlighted the multi-view architecture, in which domain engineers can still work within their specific scopes. The preliminary functionality demonstrated by the prototype shows that the engineering information can be shared across the domains consistently and efficiently. The associations among the features across the domains are modeled explicitly; example associations are established, implemented and managed in a prototype system. The dynamically generated feature parameter association map provides a graphic visualization of association information, which evolves along the lifecycle of chemical process projects. The association map can also be saved as historical records, which could be referenced in future. Based on the association information, the propagation of design changes across the domains is effectively implemented; well-informed decision-making throughout the engineering project cycle is supported.

9 Conclusions and Future Work

9.1 Conclusions

In current industrial practice, owing to the lack of an effective management method for those complex associations among engineering entities in the dynamic project environment, engineering semantics built into individual domain models are stripped off during the information transfer across domains. This drawback leads to frequent design conflicts. It is also tedious and error prone for the engineers to solve such design conflicts by reworking their design models. Ideally, when engineering semantics is shared by engineers from different disciplines, the interwoven design models will be more consistent with fewer design iterations needed. Consequently, the engineering companies could benefit by quicker and more accurate delivery. In this research, a unified interdisciplinary engineering methodology has been developed and presented to support such engineering semantics sharing across the life cycle of a chemical process project. The interdisciplinary method proposed in this research bridges the chemical process engineering and mechanical engineering domains.

The first major contribution made by this research work is the unified interdisciplinary engineering framework, under which a schema of feature models has been developed by exploring the semantic commonalities and relationships between chemical process engineering and mechanical design domains. Such a feature modeling framework consistently embeds knowledge into design models.

A schema of two categories of new features that can be identified from commonly observed chemical engineering processes, i.e., associative chemical process features and inter-domain functional features, is developed. Related to those existing product-related features in the mechanical engineering domain, the above two sets of features offer new mechanisms to support engineering semantics sharing in such a multi-disciplinary chemical process modeling system.

- The associative chemical process features provide comprehensive and flexible definitions of chemical processes in a hierarchical structure. Each chemical process feature model is capable of explicitly representing engineering templates of chemical processes and the associations across domains; their instances can be consistently used to build a knowledge-based system.
- The inter-domain functional features proposed in this research build a bridge to associate chemical process features and mechanical design features, and therefore the design intent in chemical engineering can be expressed more explicitly in a tangible object form. Such functional features benefit the synchronization between the chemical and mechanical engineering domains. Applying inter-domain functional features allows for the establishment of flexible associative relationships among interdisciplinary features, which supports convenient updates of functional mapping.

In the proposed method, the associations and cross references among different domain features are systematically managed as engineering constraints at a fine granularity. The explicit representation of engineering constraints offers quantitative evaluation capability rather than just qualitatively registering the dependencies among detailed engineering model entities. With the feature associations well-maintained along the life cycle, a feature parameter association map can be dynamically constructed whenever needed, which always reflects the most updated dependency information. The map generated in this procedure provides engineers and the change propagation algorithm a context to evaluate engineering change impacts.

Based on the dynamically generated feature parameter association map, the proposed PECSP solving method automatically searches for connected parameter and constraint nodes, and formulates CSPs with progressively expanded scope. The variables and constraints involved in the CSPs are incrementally added until the formulated CSP has a solution. The proposed method is not intended to replace the engineers but rather to assist engineers with locally-controlled and hence efficient engineering change evaluation results or conflict suggestions on engineering change implementation. Compared to solving the global problem, which will involve all feature parameters and constraints, the proposed method is scalable to handle large industrial projects, since it always attempts to solve the CSP with the smallest scope no matter how large the project is. This can save so much computation efforts that the solution suggestions can be provided to engineers in a pretty timely fashion. Also, the solution generated tends to need fewer modifications to the existing design, which will be a great advantage to engineering change implementation, such as in the design retrofit (Smith, 2005). Besides, the engineering constraint model provides an explicit and flexible expression of engineering intent, which could be a great resource for a knowledge-based system, and it can be foreseen in future the KBS could potentially be seamlessly developed to further improve the synthesis capability of design engineering systems.

After any industrial project is finished, the built-in engineering knowledge will be stored into the case database and extracted for reuse as future reference. The proposed feature library can also be enriched gradually with the growth of customized features. In turn, such historical cases and the feature library can be used to generate conceptual engineering solutions for multiple domains.

The system proposed can discover and track constraint conflicts automatically and provide the required decision-making support with the aid of computational tools. This potential stem is aimed to assist engineers rather than replace them. The compromising resolution can only be justified by the engineers. Although the consistency-maintenance and conflict-solving mechanisms still need more development, the conflict-solving mechanism has been embedded in the change propagation algorithm; so more intelligent suggestions can be provided.

9.2 Recommendation for Future Work

The authors would like to suggest the future work in the following directions.

Firstly, the investigation of fuzzy constraints should be carried out. In this research work, some constraints have not been considered yet, which are not necessarily definite in mathematical expression. Fuzzy constraints could be explored to handle such constraints. However, that will bring in the research challenge of generic representation of fuzzy constraints that also demands the reasoning process in solving fuzzy constraints. The method needs to be implemented for such more complex constraint types.

Secondly, because the design rules implemented are limited, the prototype system must be tested in real applications of industrial projects, as the overall effectiveness has yet to be proven and further refinement has to be implemented. The performance of the system can be further verified by comparing it with current industry practice with a set of typical real cases.

Thirdly, the search of the formulated CSP scope can be further enhanced based on the analysis of the established association graph. The method proposed in this research assumes that the engineering cost of changing any parameter is identical. In reality, the associated costs, such as the engineering and manufacturing costs needed to modify different parameters of equipment/components are totally different; a more intelligent cost estimation algorithm has to be developed to take this into consideration. In the future, ideally, the intelligent algorithm can lead the change propagation towards more economical modification options.

Fourthly, a more effective and efficient CSP solving algorithm can be developed in the future. Currently, the proposed method relies on the capability of commercial CSP solvers. It is expected that advanced algorithms are to be developed in future research to extend the capability in dealing with more types of constraints, and/or even hybrid types, such as continuous analytical, discrete, and fuzzy constraints in an efficient way. Besides, the solution generated from the proposed method may not be the optimum solution.

Further, more artificial intelligence research should be done to generate some compromised advice for engineers. The proposed change propagation method is supposed to find a workable plan to solve the design conflicts assuming there is at least one feasible solution. However, in the worst cases, there is no solution to even the formulated CSPs with biggest scope, i.e., design conflicts cannot be solved by propagating engineering changes. Such unsolvable conflicts will not be solved; they are left to engineers. In some other cases, although there is a solution, the snowball effect of the propagation solution may not be affordable. Then, compromises have to be made to resolve the conflicts. Some flexible constraints have to be relaxed and removed from the formulated CSPs, or design decision constraints are reevaluated to check whether these constraints can be adjusted to make the formulated CSP solvable. In this thesis, the strength of constraints has been modeled, which could provide input data to determine which constraint can be relaxed and removed from the formulated CSPs. An intelligent solving method could be developed in future to address such issue.

Bibliography

Abdul-Ghafour, S., Ghodous, P., Shariat, B., & Perna, E. (2014). Semantic interoperability of knowledge in feature-based CAD models. *Computer-Aided Design*, 56, 45-57.

Alvares, A.J., & Ferreira, J.C.E. (2008). A system for the design and manufacture of feature-based parts through the Internet. *International Journal of Advanced Manufacturing Technology*, 35, 646-664.

American Society of Mechanical Engineers (ASME). (2004). *ASME Boiler and Pressure Vessel Code, Section VIII Division 1*, New York, NY.

Anderl, R., & Mendgen, R. (1996). Modelling with constraints: theoretical foundation and application. *Computer-Aided Design*, 28(3), 155–168.

Babic, B.R., Nesic, N., & Miljkovic, Z. (2008). A review of automated feature recognition with rule-based pattern recognition. *Computers in Industry*, 59, 321-337.

Bañares-Alcántara, R. (1995). Design support systems for process engineering--i. Requirements and proposed solutions for a design process representation, *Computers & Chemical Engineering*, 19(3), 267-277.

Bañares-Alcántara, R., & King, J.M.P. (1997). Design support systems for process engineering--iii. Design rationale as a requirement for effective support. *Computers & Chemical Engineering*, 21(3), 263-276.

Bausbacher, E.F., & Hunt, R.W. (1993). *Process Plant Layout and Piping Design*. Upper Saddle River, NJ: Prentice-Hall.

Bayer, B., & Marquardt, W. (2004). Towards integrated information models for data and documents. *Chemical Engineering Science*, 28, 1249-1266.

Bettig, B., & Hoffmann, C.M. (2011). Geometric constraint solving in parametric computer-aided design. *Journal of Computing and Information Science in Engineering*, 11(2), 021001-1–021001-9.

Bettig, B., & Shah, J. (2001). Derivation of a standard set of geometric constraints for parametric modeling and data exchange. *Computer-Aided Design*, 33, 17-33.

Bidarra, R., & Bronsvoort, W.F. (2000). Semantic feature modeling. *Computer-Aided Design*, 32(3), 201-225.

Borissova, A., Fairweather, M., & Goltz, G. E. (2006). An option generation and selection methodology for process equipment selection. *Research in Engineering Design*, 17, 13-26.

Bouikni, N., Desrochers A., & Rivest, L. (2006). Geometric constraint solving in parametric computer-aided design. *Journal of Computing and Information Science in Engineering*, 6, 188-195.

Brailsford, S.C., Potts, C.N., & Smith, B.M. (1999). Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119, 557-581.

Bronsvoort, W.F., & Jansen, F.W. (1993). Feature modeling and conversion - key concepts to concurrent engineering. *Computers in industry*, 21(1), 61-86.

Bronsvoort, W.F. & Noort, A. (2004). Multiple-view feature modeling for integral product development. *Computer-Aided Design*, 36(10), 929-946.

Brown, D.C. (2002). Functional, behavioral and structural features. In *Proceedings of KIC5, 5th IFIP WG5.2 Workshop on Knowledge Intensive CAD*. USA.

Browning, T. R. (2001). Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions. *IEEE Transactions on Engineering Management*, 48(3), 292–306.

Chandrasegaran, S.K., Ramani, K., Sriram, R.D., Horváth, I., Bernard, A., Harik, R.F., & Gao, W. (2013). The evolution, challenges, and future of knowledge representation in product design systems. *Computer-Aided Design*, 45, 204-228.

Chang, A.F., Panshikanti, K., & Liu, Y.A. (2012). *Refinery Engineering – Integrated Process Modeling and Optimization*. Weinheim, Germany: Wiley-VCH.

Charpentier, J.C. (2010). Among the trends for a modern chemical engineering, the third paradigm: The time and length multi-scale approach as an efficient tool for process intensification and product design and engineering. *Chemical Engineering Research and Design*, 88, 248-254.

Chen, H. M., Hou, C.C., & Wang, Y.H. (2013). A 3D visualized expert system for maintenance and management of existing building facilities using reliability-based method. *Expert Systems with Applications*, 40, 287-299.

Chen, G., Ma, Y.-S., Thimm, G., & Tang, S.H. (2006). Associations in a unified feature modeling scheme. *ASME Transactions Journal of Computing & Information Science in Engineering*, 6(2), 114-126.

Chen, J., & Munteanu, M. (2012). Optimizing bitumen upgrading scheme - modeling and simulation approach. Paper presented at *2012 AIChE Spring Meeting and 8th Global Congress on Process Safety*, Houston, TX: American Institute of Chemical Engineers.

Chen, X., Gao, S., Yang, Y., & Zhang, S. (2012). Multi-level assembly model for top-down design of mechanical products. *Computer-Aided Design*, 44, 1033-1048.

Cherng, J.G., Shao, X.Y., Chen, Y., & Sferro, P.R. (1998). Feature-based part modeling and process planning for rapid response manufacturing. *Computers industry Engineering*, 34(2), 515-530.

Cheng, X., Wang, Y., & Liu, H. (2009). Study on optimal regulation modelling of reservoir discharge. In C. Zhang & H. Tang (Ed.), *Advances in Water Resources and Hydraulic Engineering - Proceedings of 16th IAHR-APD Congress and 3rd Symposium of IAHR-ISHS* (pp. 89-115). Beijing, China: Springer.

Chu, X., Tang, C., Xue, D., Su, Y., & Sun X. (2012). Identification of machining features based on available resources of cutting tools. *International Journal of Production Research*, 50(15), 4141-4157.

Clarkson, P.J., Simons, C., & Eckert, C.M., 2004. Predicting change propagation in complex design. *Journal of Mechanical Design*, 126 (5), 765-797.

Couper, J.R., Penney, W.R., Fair, J.R., & Walas, S.M. (2010). *Chemical process equipment selection and design* (3rd ed.). Oxford, UK: Elsevier.

Cruze, J., & Barahona, P., 2005. Constraint reasoning in deep biomedical models. *Artificial Intelligence in Medicine*, 34, 77-88.

Deng, Y.M., Britton, G.A., Lam, Y.C., Tor, S.B., & Ma, Y.S. (2002). Feature-based CAD-CAE integration model for injection-moulded product design. *International Journal of Production Research*, 40(15), 3737-3750.

Dechter, R. (2003). *Constraint Processing*. San Francisco, CA: Morgan Kaufmann Publishers.

Douglas, J.M. (1988). *Conceptual Design of Chemical Processes*. New York, NY: McGraw-Hill.

Eden, M.R., Jørgensen, S.B., Gani, R., & El-Halwagi, M.M. (2004). A novel framework for simultaneous separation process and product design. *Chemical Engineering and Processing*, 43, 595-608.

Erden, M.S., Komoto, H., Van Beek, T.J., D'Amelio, V., Echavarria, E., & Tomiyama, T. (2008). A review of function modeling: approaches and applications, Artificial Intelligence for Engineering Design, *Analysis and Manufacturing*, 22, 147-169.

Gernaey, K.V., & Gani, R. (2010). A model-based systems approach to pharmaceutical product-process design and analysis. *Chemical Engineering and Processing*, 65, 5757-5769.

Gelle, E., & Faltings, B. (2003). Solving mixed and conditional constraint satisfaction problems. *Constraints*, 8, 107-141.

Gray, M.R. (1994). *Upgrading Petroleum Residues and Heavy Oils*. New York, NY: Marcel Dekker.

Granvilliers, L., & Benhamou, F. (2006). Algorithm 852: RealPaver: an interval solver using constraint satisfaction techniques. *ACM Transactions on Mathematical Software*, 32(1), 138-156.

Guirardello, R., & Swaney R.E. (2005). Optimization of process plant layout with pipe routing. *Computers and Chemical Engineering*, 30, 99-114.

Hamraz, B., Caldwell, N.H.M., & Clarkson, P.J. (2013). A matrix-calculation-based algorithm for numerical change propagation analysis. *IEEE Transactions on Engineering Management*, 35, 692–708.

Hayasi, M.T., & Asiabanpour B. (2009). Extraction of manufacturing information from design-by-feature solid model through feature recognition. *International Journal of Advanced Manufacturing Technology*, 44(11–12), 1191-1203.

Hoos, H.H., & Tsang, E. (2006). Local search methods. In F. Rossi, P. van Beek, & T. Walsh (Ed.), *Handbook of Constraint Programming* (pp. 135-167). Oxford, UK: Elsevier.

Intergraph (2014). SmartPlant 3D. Available from: http://www.intergraph.com/.

Jarratt, T.A. W., Eckert, C.M., Caldwell, N.H.M., & Clarkson, P.J. (2011). Engineering change: an overview and perspective on the literature. *Research in Engineering Design*, 22, 103-124.

Jarratt, T.A.W., Eckert, C.M., & Clarkson, P.J. (2004). Engineering change. In *Design process improvement- A review of current practice*. London, UK: Springer.

Jarullah, A.T., Mujtaba, I.M., & Wood, A.S. (2012). Improving fuel quality by whole crude oil hydrotreating: a kinetic model for hydrodeasphaltenization in a trickle bed reactor. *Applied Energy*, 94, 182-191.

Jaworski, Z., & Zakrzewska, B. (2011). Towards multi-scale modeling in product engineering. *Computers and Chemical Engineering*, 35, 434-445.

Keller, R., Eckert, C.M., & Clarkson, P.J. (2009). Using an engineering change methodology to support conceptual design. *Journal of Engineering Design*, 20(6), 571-587.

Klatt, K.U., & Marquardt, W. (2009). Perspectives for process systems engineering – personal views from academia and industry, *Computers and Chemical Engineering* , 33, 536-550.

Koh, E.C.Y., Caldwell, N.H.M., & Clarkson, P.J. (2012). A method to assess the effects of engineering change propagation. *Research in Engineering Design*, 23, 329-351.

Lam, S.M., & Wong, T.N. (2000). Recognition of machining features - a hybrid approach. *International Journal of Production Research*, 38(17), 4301-4316.

Lee, S.H. (2005). A CAD–CAE integration approach using feature-based multi-resolution and multi-abstraction modelling techniques. *Computer-Aided Design*, 37, 941-955.

Lee, S.H. (2009). Feature-based non-manifold modeling system to integrate design and analysis of injection molding products. Journal of *Mechanical Science and Technology*, 23, 1331-1341.

Louhichi, B., & Rivest, L. (2014). Maintaining consistency between CAD elements in collaborative design using association management and propagation. *Computers in Industry*, 65, 124-135.

Luyben, W.L. (2010). Heuristic design of reaction/separation process. *Industrial & Engineering Chemistry Research*, 49, 11564-11571.

Ma, Y.-S., & Bong, C.H. (2010). Fine grain associative feature reasoning in collaborative engineering. *International Journal of Computer Applications in Technology*, 37(3–4), 210-216.

Ma, Y.-S., Britton, G.A., Tor, S.B., & Jin, L.Y. (2007). Associative assembly design features: concept, implementation and application. *International Journal of Advanced Manufacturing Technology*, 32(5-6), 434-444.

Ma, Y.-S., Chen, G., & Thimm G. (2008). Paradigm shift: unified and associative feature-based concurrent and collaborative engineering. *Journal of Intelligent Manufacturing, special issue on Advanced Technologies for Collaborative Manufacturing*, 19(6), 626-641.

Ma, Y.-S., Chen, G., & Thimm G. (2009). Fine Grain Feature Associations in Collaborative Design and Manufacturing - A New Modelling Approach. *Collaborative Design and Planning for Digital Manufacturing*, Springer, 71-97.

Ma, Y.-S., & Tong, T. (2003). Associative feature modeling for concurrent engineering integration. *Computers in Industry*, 51(1), 51-71.

Mahalec, V., & Motard, R.L. (1977). Procedures for the initial design of chemical processing systems. *Computers and Chemical Engineering*, 1, 57-68.

Marquardt, W., & Nagl, M. (2004). Workflow and information centered support of design processes—the IMPROVE perspective. *Computers & Chemical Engineering*, 29(1), 65-82.

McCabe, W.L., Smith, J.C., & Harriott, P. (2005). *Unit Operations of Chemical Engineering*, 7th edition. Boston, MA: McGraw-Hill.

Meyers, R.A. (2004). *Handbook of Petroleum Refining Processes* (3rd ed.). New York, NY: McGraw-Hill.

Miller, D.C., & Davis, J.F. (2000). Process design decision support system for developing process chemistry, *Ind. Eng. Chem. Res*, 39(8), 2954-2969.

Moggridge, G.D., & Cussler, E.L. (2010). An introduction to chemical product design. *Chemical engineering research and design,* 78(A1), 5-11.

Mokhtar, A., & Xu, X. (2011). Machining precedence of 2½D interacting features in a feature-based data model. *Journal of Intelligent Manufacturing*, 22(2), 145-161.

Morales-Rodríguez, R., Gani, R., Déchelotte, S., Vacher, A., & Baudouin, O. (2008). Use of CAPE-OPEN standards in the interoperability between modelling tools (MoT) and process simulators (Simulis® Thermodynamics and ProSimPlus). *Chemical Engineering Research and Design*, 86, 823-833.

Morbach, J., Wiesner, A., & Marquardt, W. (2009). OntoCAPE—a (re)usable ontology for computer-aided process engineering. *Computers and Chemical Engineering*, 33, 1546-1556.

Naranje, V., & Kumar, S. (2014). A knowledge based system for automated design of deep drawing die for axisymmetric parts. *Expert Systems with Applications*, 41, 1419-1431.

Ouertani, M.Z. (2008). Supporting conflict management in collaborative design: An approach to assess engineering change impacts. *Computers in Industry*, 59, 882-893.

Ouertani, M.Z., & Gzara, L. (2008). Tracking product specification dependencies in collaborative design for conflict management. *Computer-Aided Design*, 40, 828-837.

Pang, W. (1997). *Constraint structure in constraint satisfaction problems* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses Global.

Pelleau, M., Truchet, C. & Benhamou, F. (2014). The octagon abstract domain for continuous constraints. *Constraints*, 19, 309-37.

Persson, L.G., Santos, F.B., Tavares, C.A.C., & Andrade, A.E. (2009). Methodology of pipe and equipment layout for on-shore oil & gas industry. *10th International Symposium on Process Systems Engineering - PSE2009*.

Qureshi, A.J., Dantan, J.Y., Bruyere, J., & Bigo, R. (2010). Set based robust design of mechanical systems using the quantifier constraint satisfaction algorithm. *Engineering Applications of Artificial Intelligence*, 23, 1173-1186.

Razgon, I.. (2013). Complexity analysis of heuristic CSP search algorithms. In *Recent Advances in Constraints*. London: Springer.

Rouibah, K., & Caskey, K.R. (2003). Change management in concurrent engineering from a parameter perspective. *Computers in Industry*, 50, 15-34.

Russel S., & Norvig, P. (2010). *Artificial intelligence: a modern approach* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Salehi, V., & McMahon, C. (2011) Development and application of an integrated approach for parametric associative cad design in an industrial context, *Computer-Aided Design and Applications*, 8(2), 225-236.

Schichl, H., Neumaier A., Markót, M.C., & Domes, F. (2013). On solving mixed-integer constraint satisfaction problems with unbounded variables. In C. Gomes and M. Sellmann (Eds.), *CPAIOR 2013, LNCS 7874* (pp. 216-233). Berlin, Germany: Springer.

Schmidt-Traub, H., Köster, M., Holtkötter, T., & Nipper, N. (1998). Conceptual plant layout. *Computers and Chemical Engineering*, 22, S499-S504.

Schneider, R., & Marquardt, W. (2002). Information technology support in the chemical process design life cycle. *Chemical Engineering Science*, 57(10), 1763-1792.

Schulte, C., Tack., G., & Lagerkvist, M.Z. (2010). *Modeling and programming with Gecode.* Retrieved from http://www.gecode.org/doc-latest/MPG.pdf.

Shah, J.J. (1991). Assessment of features technology. *Computer-Aided Design*, 23(5), 331-343.

Shah, J.J., & Mäntylä, M. (1995). *Parametric and feature-based CAD/CAM*. New York: John Wiley & Sons.

Siirola, J.J., Powers, G.J., & Rudd, D.F. (1971). Synthesis of system designs: iii. Toward a process concept generator. *AIChe Journal*, 17(3), 677-682.

Smith, R. (2005). Chemical process design and integration. Chichester, UK: John Wiley & Sons.

Speight, J.G. (2011). *The Refinery of the Future*. Oxford, UK: Elsevier.

Stephanopoulos, G., & Reklaitis, G.V. (2011). Process systems engineering: from solvay to modern bio- and nanotechnology. A history of development, successes and prospects for the future. *Chemical Engineering Science*, 66, 4272-4306.

Stützle, W. (1998). *Local search algorithms for combinatorial problems-Analysis, Improvements, and New Applications* (Doctoral dissertation, Technische Universität Darmstadt). Retrieved from http://iridia.ulb.ac.be/~stuetzle/publications/Thesis.ThomasStuetzle.pdf.

Sudarsan, R., Fenves, S.J., Sriram, R.D., & Wang, F. (2005). A product information modeling framework for product lifecycle management. *Computer-Aided Design*, 37, 1399-1411.

Syaimak, A.S., & Axinte, D.A. (2009). An approach of using primitive feature analysis in manufacturability analysis systems for micro-milling/drilling. *International Journal of Computer Integrated Manufacturing*, 22(8), 727-744.

Tang, S.H., Chen, G., & Ma, Y.-S. (2013). Fundamental concepts of generic Features. In Y. Ma (Ed.), *Semantic Modeling and Interoperability in Product and Process Engineering* (pp. 89-115). London, UK: Springer.

Tessier, S., & Wang, Y. (2013). Ontology-based feature mapping and verification between CAD systems. *Advanced Engineering Informatics*, 27, 76-92.

Towler, G., & Sinnott, R.K. (2013). *Chemical Engineering Design - Principles, Practice and Economics of Plant and Process Design* (2nd ed.). Waltham, MA: Elsevier.

Trombettoni, G., Papegay, Y., Chabert G., & Pourtallier, O. (2010). A box-consistency contractor based on extremal functions. In D. Cohen (Ed.), *Principles and Practice of Constraint Programming - CP 2010* (pp. 491–498). Berlin, Germany: Springer.

Van Beek, P. (2006). Backtracking search algorithms. In F. Rossi, P. van Beek, & T. Walsh (Ed.), *Handbook of Constraint Programming* (pp. 135-167). Oxford, UK: Elsevier.

Van der Meiden, H.A., & Bronsvoort, W.F. (2006). A constructive approach to calculate parameter ranges for systems of geometric constraints. *Computer-Aided Design*, 38, 275–283.

Van Hentenryck, P., Michel, L., & Deville, Y. (1997). *Numerica : a modeling language for global optimization*. Cambridge, MA: The MIT Press.

Verhagen, W.J.C., Bermell-Garcia, P., Dijk, R.E.C., & Curran, R. (2012). A critical review of Knowledge-Based Engineering: An identification of research challenges. *Advanced Engineering Informatics*, 26, 5-15.

Verma, A.K., & Rajotia, S. (2010). A review of machining feature recognition methodologies. *International Journal of Computer Integrated Manufacturing*, 23(4), 353-368.

Vijaykumar, G., & Chakrabarti, A. (2008). Understanding the knowledge needs of designers during design process in industry. *ASME Transactions Journal of Computing and Information Science in Engineering*, 8(1), 0110041–0110049.

Vu, X.-H., Schichl, H., & Sam-Haroud, D. (2009). Interval propagation and search on directed acyclic graphs for numerical constraint solving. *Journal of Global Optimization*, 45, 499-531.

Wiesner, A., Morbach, J., & Marquardt, W. (2011). Information integration in chemical process engineering based on semantic technologies. *Computers and Chemical Engineering*, 35, 692-708.

Xie, Y., Liu, J., Liu H., & Ma, Y. (2013). Features and interoperability of computer aided engineering systems. In Y. Ma (Ed.), *Semantic Modeling and Interoperability in Product and Process Engineering* (pp. 143-191). London, UK: Springer.

Xu, Z., Zhang, J., Li, Y., Jiang, S., & Sun, Y. (2013). Product modeling framework based on interaction feature pair. *Computer-Aided Design*, 45, 1591-1603.

Xue, D., & Yang, H. (2004). Product A concurrent engineering-oriented design database representation mode. *Computer-Aided Design*, 45, 947-965.

Yeung, C.L., Cheung, C.F., Wang, W.M. & Tsui, E. (2014). A knowledge extraction and representation system for narrative analysis in the construction industry. *Expert Systems with Applications*, 41, 5710-5722.

Zheng, Y., Taib, J.M., & Tap, M.M. (2012). Decomposition of interacting machining features based on the reasoning on the design features. *International Journal of Advanced Manufacturing Technology*, 58(1–4), 359-377.

Zhou, X., Qiu, Y., Hua, G., Wang, H., & Ruan, X. (2007). A feasible approach to the integration of CAD and CAPP. *Computer-Aided Design*, 39, 324-338.

Zitney, S.E. (2010). Process/equipment co-simulation for design and analysis of advanced energy systems. *Computers and Chemical Engineering*, 34, 1532-1542.