A METHODOLOGY FOR ENGINEERING DESIGN CHANGE ANALYSIS USING SYSTEM MODELLING AND KNOWLEDGE MANAGEMENT TECHNOLOGIES

GENYUAN FEI

A thesis submitted in partial fulfilment of the

requirements of the University of Greenwich

for the Degree of Doctor of Philosophy

May 2011

DECLARATION

"I certify that this work has not been accepted in substance for any degree, and is not concurrently being submitted for any degree other than that of Doctor of Philosophy being studied at the University of Greenwich. I also declare that this work is the result of my own investigations except where otherwise identified by references and that I have not plagiarised the work of others."

Genyuan Fei

Candidate

Supported By:

Professor James Gao

Supervisor

ACKNOWLEDGEMENTS

This work would not have been possible without the constant guidance, constructive feedback, and help of my supervisor, Professor James Gao, to whom I am deeply grateful. I thank him for being my supervisor and giving me freedom to pursuit my own research interest. I also give my thanks to my second supervisor, Dr. Dele Owodunni, for his constant support and help.

I express my thanks to the School of Engineering and the University of Greenwich for providing me with a productive work environment and supporting my study with a research bursary.

I thank my parents and my sister for always having confidence in me and for their love and encourage. My special thanks to my wife, Yuehua, for everything she did for me. She has been my source of inspiration and strength. Her love and care have kept me afloat through tough time.

There are many others who had helped me along the way, and I would like express my thanks to all of them.

LIST OF PUBLICATIONS

Fei, G., Gao, J., Owodunni, D. and Tang, X., 2011. A Model-driven and Knowledge-based Methodology for Engineering Design Change Management. *Computer-Aided Design and Applications*, 8(3), pp. 373-382.

Fei, G., Gao, J., Owodunni, D. and Tang, X., 2011. A Method for Engineering Design Change Analysis Using System Modelling and Knowledge Management Techniques. *International Journal of Computer Integrated Manufacturing*, 24(6), pp. 535-551.

Fei, G., Gao, J., Owodunni, D. and Tang, X., 2010. A Methodology for Engineering Design Change Management Using Modelling and Problem Solving Techniques. In: HINDUJA, S. and LI, L., ed. *The 36th International MATADOR Conference*, Manchester, pp. 179-182.

Fei, G., Gao, J., Owodunni, D. and Tang, X. Q., 2009. A TRIZ based methodology for the analysis of the coupling problems in complex engineering design. *The 19th CIRP Design Conference - Competitive Design.* Cranfield University: Cranfield University Press, pp. 285-292.

CONTENTS

FIGURES

ABSTRACT

In the current fiercely competitive market, engineering design change management in manufacturing companies is a critical factor for business success. Recognised as inevitable in product development, engineering changes may significantly influence lead time, development cost and product quality of new product development. It has been recognised that the earlier change issues are addressed, the greater product lifecycle costs can be saved.

In this thesis, three main industrial requirements have been identified in engineering design change management, including: (i) a lack of formal methods to analyse the impacts of design changes on both functional requirements and physical components; (ii) a lack of methods to trace design change propagations; (iii) a lack of methods for conflict resolving in design change management. It is also identified that there is a lack of systematic method for reusing design knowledge to solve design conflicts. The literature review carried out in this project also confirms that there were no unified and systematic methods proposed to meet these industrial requirements.

This thesis reports a methodology and tool to meet these requirements and help designers trace, analyse and evaluate engineering changes occurring in the product design phase. A modelling method is employed to enhance the traceability of potential design changes occurred between the functional requirements domain and physical structure domain of design. Based on functional and physical models, a matrix-based method is developed to analyse change propagations between components and help find out design conflicts arising from design changes. A knowledge based method has been proposed to resolve design conflicts by reusing previous design change knowledge. A web-based distributed system has been developed to implement the proposed methodology. An engineering design change example from the collaborating company has been used in a case study to help understand the methodology and prove its usefulness.

CHAPTER 1 INTRODUCTION

In this chapter, the general trends of the manufacturing industry are discussed and challenges in the current globalised environment are identified. New product development as a sustainable measure to drive the growth of the manufacturing industry is stressed. As an important part of new product development, engineering design change management is discussed. Current research and practices in engineering design change management are briefly introduced and the research gaps are indicated. Then the aim and the objectives of this research are stated. Finally, the structure of the thesis is described.

1.1 Trends and Challenges of the Manufacturing Industry

Since the Industrial Revolution started in the United Kingdom in the 18^{th} century, countries which gained the strongest manufacturing industry normally have had the most powerful economy. This trend had been witnessed with the power shifting from the United Kingdom to the United States in the past more than one century. And now, emerging economies like China, India and Brazil are on their way to take over the positions by the continuously booming manufacturing industries in their countries. The manufacturing industry in the developed countries has been seen in decline for many years (United Nations Conference on Trade and Development, 2010). Taking the UK's economy for example, from 1997 to 2009, the proportion of manufacturing contributing to the country's gross domestic product (GDP) has steadily declined from 22% to just over 11% (British Department for Business Innovation & Skills, 2010). The main reason behind this situation is that the emerging economies with lower labour costs have started taking over the market share of mass production and low value-added manufactured goods.

However, even if in developed countries like the UK, the manufacturing industry has been declining in terms of its proportion of contribution to the country's income, it is still a critical constituent of the whole economy (National Statistics, 2006). Still taking the UK for example, the manufacturing industry is still the third largest sector which accounts for 12% of the country's economic output and employs around 2.5 million workers (Mellows-Facer, 2010). Although declining in the mass production and low value-added part of the manufacturing business, developed countries are still the leaders in the high value-added part and there is no sign showing that their share of high value-added manufacturing is declining. From the statistical figures from the OECD STAN Bilateral Trade Data in 2008, it is seen that nearly 65% of manufactured exports of the UK are from high and medium-high technology. Extracted from the latest government document in 2010, the UK government stressed the importance of the manufacturing industry as a part of a healthy and balanced economy and has also planned to boost high-tech advanced manufacturing in the next few years (Mellows-

Facer, 2010). The UK government's plan has also been echoed by PricewaterhouseCoopers who is the world's second largest professional service company and one of the so-called Big-Four accountancy companies. In a report from them, they have analysed the manufacturing sector in the UK and given their recommendations to the sector (Pricewaterhousecoopers, 2009). In one of their recommendations, they suggested that the UK manufacturing industry focuses more on knowledge-based innovative activities rather than production since production always goes to places with lower labour costs. It is suggested to re-apply their core knowledge of the industry to new products rather than the out-dated and commoditised product range, to focus more on research and development, to dedicate to product quality, reliability and responsiveness, and to concentrate on customisation to provide innovative products.

On the other side of the table of the global manufacturing industry, developing countries with emerging economies have taken a significant market share of manufactured goods production. Very notable, in only nearly two decades China has transformed from a traditionally agricultural country into an industrial country with the second largest manufacturing output behind the US. The manufacturing output of China makes up about 42.7% of its GDP in 2008 (British Department for Business Innovation & Skills, 2010). And recently, it outperformed Japan and became the world's second largest economy thanks to its strong growth in manufacturing. But it has been recognised that low-cost led growth is not sustainable, since when the country is gradually developed, the labour costs will increase as well. Manufacturing companies will then look for other places with lower labour costs. Still taking China as an example, there are many reports recently showing that many companies are considering to move or have already moved their plants in China to Vietnam where the labour costs is even lower compared to the southern part of China (Mason, 2010; Folkmanis and Giang, 2010). Therefore, even developing countries with lower labour costs at the moment also need to consider new product development so that their growth can be sustainable. In a recent report, researchers found that China has overtaken United States in 2009 and become the No. 1 investor in renewable energy technologies which is seen as an important growing business in manufacturing industry (Black, 2010).

It can be clearly seen from these evidences that the manufacturing industry, which produces tangible goods and intangible services, is still the mainstream of the world's economy. Although low value-added manufacturing activities always keep moving from countries with high labour costs to countries with lower labour costs, success in new product development is the key point to sustainable growth of the sector in the long run. This is widely recognised in both developed and developing countries. Therefore, research in new product development can be significantly rewarding in terms of its contribution to the world's manufacturing industry.

1.2 Engineering Change Management and its Issues

In order to survive and grow in such an increasingly competitive business environment, companies struggle to turn their market places from "Red Ocean" to "Blue Ocean" (Kim and Mauborgne, 2005). In the "Red Ocean", companies have to seek to reduce costs and profits in order to survive in a matured market with fierce competition. While the essence of "Blue Ocean" strategy is the success of new products, which leads companies to a non-competitive or less competitive market place and enlarges the profit space. In other words, new product development is not only the critical linkage between a business organisation and its market, but also fundamental to business success.

The importance of new product development to business success is widely recognised by researchers and industrial practitioners. Fitzsimmons et al. (1991) stated that firms that fail to manage their product development activities strategically are not only running their business from a position of disadvantage but are risking their future. It was also reported that almost 40 percent of sales came from new products in America (Cooper and Kleinschmidt, 1995). In another way, Whitney (1988) stated that a new product development environment is a strategic business activity either by intent or by default.

Engineering change management is seen as an important issue in new product development. As a successful new product development project, it is required to effectively and efficiently respond to any changes during the product development.

These changes could be caused by customer requirements, technical requirements, product enhancement, quality problem solving, government policies, changes in the supply chain, and so on. A study (Stanev et al., 2008) shows that companies have 220 changes requests per month on average nowadays, while 22% of them are manufacturing related changes. In the automotive industry, the figures are even higher approximately 425 changes per month and 35% of them in manufacturing area. It is also found that from 1994 to 2005 the change processing time had tripled.

Engineering changes have been recognised as inevitable in complex engineering product development (Huang et al., 2003; Palani Rajan et al., 2005; Keller et al., 2009; Pikosz and Malmqvist, 1998). They have great influences on downstream development and production activities, thus making product development very costly and time consuming (Huang et al., 2003). Some researchers believe that engineering changes can determine 70% - 80% of the final cost of the product (Mcintosh, 1995). Therefore, it is critical to keep them under control. Engineering changes have also been recognised as a source of innovation and creativity which can facilitate evolutions of products and technologies (Balogun and Jenkins, 2003; Jarratt et al., 2003; Eckert et al., 2004). From this perspective, knowledge acquired from engineering changes is also very useful to product development in the long term. Despite of the different perspectives, both of the two arguments reflect the importance of Engineering Change Management (ECM) in product development.

In the past decade, a lot of research and development work have been done in engineering change management regarding computerising traditional paper-based engineering change processes (Huang et al., 2001), improving communicating methods between engineers (Shiau and Wee, 2008) and clarifying knock-on change effects between components (Eckert et al., 2006; Clarkson et al., 2004). Most early research shows the efforts made in dealing with engineering changes in the manufacturing process. Recently, changes happening in the critical stage of product development, the product design stage, have been emphasised. It is recognised that the design stage of product development could determine the largest cost savings during the product life cycle. This means changes happening at the design stage would have a greater impact than those happening in the manufacturing phase.

Although engineering change has been studied for many years, its definition varies according to statements of different researchers. Wright (1997) defined engineering change as modification to a component of a product before it goes into production. Some researchers agree that engineering change is modification to dimensions, fits, forms, functions and materials to a product or components after the product design is released (Huang et al., 2003; Kocar and Akgunduz, 2010). Whilst some other researchers view engineering changes as changes occurring in a wider range from customer requirements to product in use (Pikosz and Malmqvist, 1998; Eckert et al., 2004).

While research focuses have been shifting over time, the scope of research on engineering change has been widened. The initial motivation of studying ECM was to avoid engineering changes during the manufacturing process due to the adverse effects they cause. The adverse effects caused in terms of delivery time, developing cost and product quality are noticeable but very difficult to estimate (Huang et al., 2003). Later on, people realised that engineering changes are actually inevitable. Therefore, researchers have turned to finding out how engineering changes go on and what kind of impact they may cause (Clarkson et al., 2004; Kocar and Akgunduz, 2010; Ouertani, 2008). Recently, many researchers found the benefit of engineering changes to innovation and creativity, which can enhance the competitiveness of companies. Thus some researchers have started to study engineering changes from perspectives of knowledge management and knowledge reuse (Balogun and Jenkins, 2003; Palani Rajan et al., 2005; Lee et al., 2006; Keese et al., 2009).

The process of organising engineering change activities have also been explored in the past decade, in order to find most efficient and effective approach of engineering change management. A general process of engineering change has been proposed by Clarkson and Eckert (2005). This process includes six steps, namely engineering change request, possible solution identification, risk/impact assessment, solution selection and approval, solution implementation, and change process review. Although the general process of engineering describes a reasonable approach to addressing change issues in product development, in reality, different companies have quite different processes to deal with engineering changes in order to fit their specific organisational and production requirements (Pikosz and Malmqvist, 1998; Eckert et al., 2004; Huang et al., 2003).

Based on previous research reviewed, some gaps in engineering change management have been identified.

Firstly, changes of functional requirements should be considered together with changes in the physical domain in the design phase. For example, according to the domain definitions of the product design stage in the theory of Axiomatic Design (Suh, 2001), when changes in physical components are considered, the changes in the functional domain and their effects on the physical domain have not been considered.

Secondly, there is a lack of consideration of the impact of change solutions on the change propagation analysis. A lot of efforts have been made to predicting change propagations with predefined component dependencies. However, specific solutions for change requests may dramatically change predefined interacting relationships between components, which may make predictions of later changes fail.

Thirdly, there is a lack of tools to help engineering designers reuse knowledge from previous cases regarding design change management in industry. In many design change cases, technical solutions for a design change request, or say for some similar design change requests, may have been re-developed on many other occasions. That may be because experience or technical solutions from previous design change cases have not been formalised and shared effectively.

1.3 Research Aim and Objectives

The aim of this research is to provide a method to analyse engineering change in the product design phase, resolve design conflicts arising from engineering change and evaluate change solutions.

The research objectives include:

- To investigate previous and current research and development work in engineering design change management and identify research gaps and industrial requirements;

- To propose a framework for design change management in and between functional and physical domains of product design;

- To propose a model driven method to analyse change propagation and identify design conflicts arising from change propagation;

- To propose a knowledge-based method to solve design conflicts by reusing previous design cases;

- To propose a method to evaluate and prioritise change propagation paths in terms of impact on project success;

- To develop a prototype system implement the proposed methodology; and

- To develop a case study for validating the system using industrial applications and interviewing users;

1.4 Research Scope and Novelty

The research scope of this project is confined to the design phase of electromechanical product development. The design phase in this research is not a particular period of a sequential product development process which is dedicated to pure design activities. It is rather a type of development activities related to the product design in an iterative product development process. The research is focused on electromechanical products that have a certain degree of complexity. It is also limited to new product development projects that normally commit incremental improvements to existing products. The potential users targeted in this research are engineering designers and design managers.

The research carried out in this thesis is based on in-depth investigations in the academic area and the industrial area. Research gaps are found from comprehensive academic literature review in Chapter 2. Industrial requirements are identified in close observations in the investigated company which is a typical small and medium enterprise focused on design and development. In order to address these research gaps and meet the industrial requirements, the proposed methodology is considered as novel for engineering design change analysis in terms of the following aspects:

Firstly, the functional domain of product design is systematically considered in engineering design change analysis, which enhances the completeness and refines the results of the change propagation analysis.

Secondly, the proposed approach to analysing design change propagation and identifying design conflicts is devised on a causal basis, which is different from many other methods that are based on estimated change possibilities. The causal approach enables the analytical process to be fact-based, and therefore the results can be more reliable.

Thirdly, it provides a systematic and easy-to-use method for engineering designers to formalise design cases and store them as knowledge, which is based on the pre-defined domain ontology.

Fourthly, the approach to finding solutions for design conflicts is devised based on a more practical and easier to use method which compares semantic similarities between formalised design conflicts with knowledge entries in the knowledge repository.

Finally, solutions for design changes are evaluated from the perspective of project success. The technical feasibilities are addressed by design change propagation analysis method, so the solution can be evaluated from a higher level of project management.

1.5 Introduction to the Collaborating Company

As one of the important parts of research in engineering, industrial investigation was carried out in a company in the manufacturing industry. In the investigation, data was collected and industrial requirements were identified. Therefore, the methodology is devised to meet the industrial requirements and benefit the investigated company and other companies in the same or similar sector.

The company investigated is a research and development oriented manufacturing company in the green energy industry. Vensys Energy AG is a Small and Medium Enterprise (SME) focused on design and development of gearless wind turbines. They design gearless wind turbines and sell production and deployment licenses to wind energy manufacturing companies. It also has strong international and distributed collaborations. They have worked very closely with local and international manufacturers and wind energy companies to produce and install wind turbines across the world.

The wind turbine company is a typical design and development company dealing with innovative and promising products. They have two product lines at the moment. One is the old 1.5 MWs wind turbine which has been in production status for a couple of years and still in continuous improvement. Most wind turbines that they have installed so far are based on this solution. The other one is the enhanced 2.5 MWs wind turbine, which is under development. As of May 2010, there are nearly 2000 wind turbines based on their solutions deployed in Europe, Asia and America.

The complexity of a wind turbine makes engineering design change management very difficult in Vensys. A wind turbine is a very complex machine with more than 3000 parts, which is developed with a wide range of technologies in multiple disciplines. A single failure of a wind turbine could cost a wind power company millions of dollars because of the losses of electricity companies. One major problem they are facing in their engineering design change management process is that there is no effective method to analyse and resolve change propagations with consideration of functional requirements and on physical components. They also find that it is difficult to resolve various design conflicts emerging from change propagation analysis, since it normally requires engineers from different backgrounds to work together. But often that requirement is hard to meet. The other major problem is that most knowledge generated from product design and development has not been systematically managed, and there is a lack of method to effectively and efficiently retrieve needed knowledge to solve design problems.

The proposed methodology, together with the developed software system, can benefit the investigated company and other similar companies in the engineering design and development domain in that:

- It makes it easier for engineering designers to intuitively model product designs and track change propagations between functional requirements and physical components;

- It can help designers identify and formalise design conflicts in the change propagation analysis process based on a causal approach;

- It provides an approach to formalising design cases and storing them in a knowledge repository, thus companies can transform and preserve their design knowledge as an asset in a systematic way;

- It can help designers automatically find reference solutions for design conflicts from the knowledge repository using a reasoning mechanism which is based on comparison of semantic similarities; and

- It can help designers evaluate candidate change solutions from the perspective of project success by using information extracted from other enterprise systems.

1.6 Research Approach and Methods Used

Throughout the research process, there are six research stages, including the research scoping stage, the literature review stage, the industrial investigation stage, the methodology development stage, the software tool development stage and the system test and validation stage. In each stage, there are different research methods and techniques employed to fulfil the tasks. Figure 1-1 depicts the research process and relevant research methods and techniques that are used in each stage.

1.6.1 Research Scoping

As the beginning stage of the research process, research scoping is aimed to define the topic of the research based on the interest and background of the researcher, and formulating the initial research objectives. With regards to the particular research documented in this thesis, a general and wide-ranging literature review has been carried out in research domains like new product development, project management, manufacturing technologies, and knowledge management. Various sources and materials are used at this stage, including academic books, influential journals and theses in these domains, internet search engines (Google scholar), and so on. As the results from this stage of the research, a general definition of the research topic is set with carefully selected and defined keywords. Also the initial research objectives are formulated.

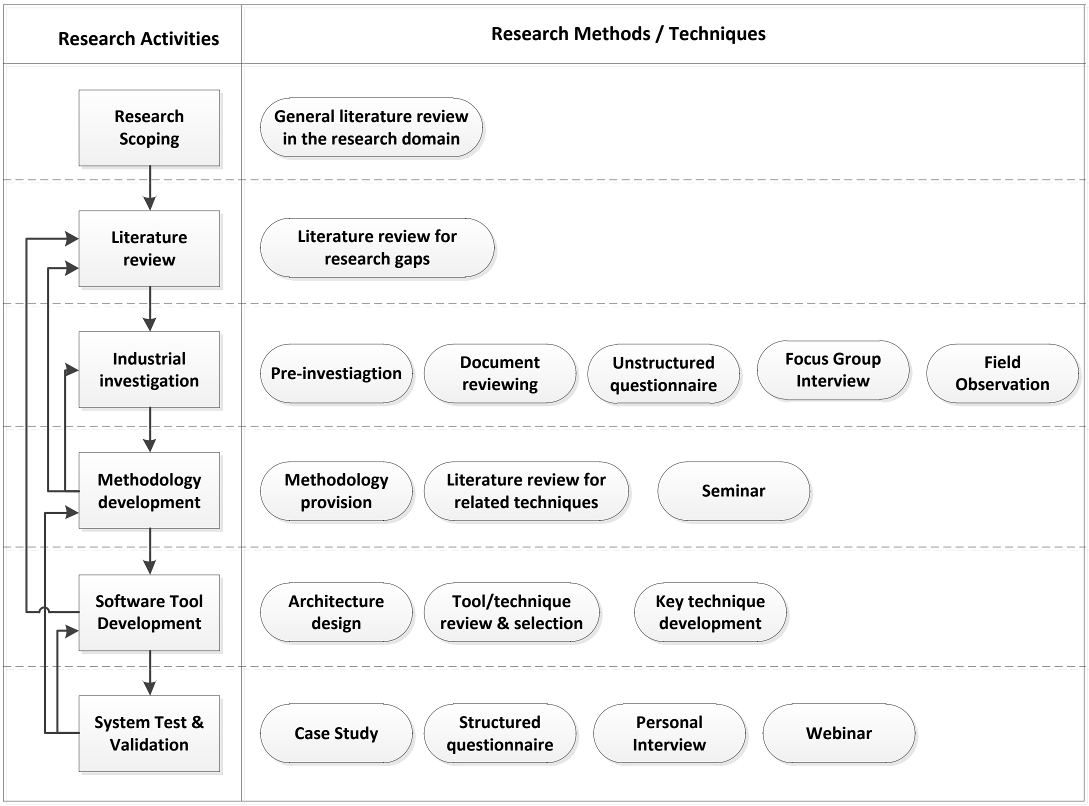


Figure 1-1 Research stages and methods

1.6.2 Literature Review

With consideration of the defined keywords and the initial research objectives, literature review is carried out in this stage where useful published articles and books are selected for reference. The aim of literature review in this stage is to find and learn research work carried out in the research domain which is defined by the research keywords and research objectives. Research gaps are identified during the literature review and the initial objectives are revised according to these research gaps. Libraries, e-libraries and digital journal hosting services have been heavily used in this stage. The keywords are searched in library catalogues and journal hosting websites. Selected books and published articles are screened at first in order to exclude those using similar keywords but not actually in the same research domain. The rest of them are carefully read and their results are analysed. Similar research work from different researchers is compared and classified.

1.6.3 Industrial Investigation

The main purposes of industrial investigations are to understand the way companies deal with engineering changes in the product design and development stage, and also identify challenges their engineers currently have. As one of the objectives of the research project, the research results intend to help them cope with the challenges.

One of the problems with a lot of research in engineering is that researchers want to take more resources to do their investigation and dig as deeply as possible, while people in industry do not have time to spend with the researchers because of tight deadlines to meet. In order to get sufficient information while taking less time from engineers, four main methods which have been used in the industrial investigation for this research are described below.

Pre-investigation is one of the important measures to conduct efficient industrial investigations without taking too much time from the companies. Web searching is employed as an effective and efficient way of pre-investigation. Not like in the old days, most companies have their own websites and online publications to introduce and advertise their companies and products nowadays. At this stage, the website of the company is explored to get an overview of their business area, product lines, achievements and collaboration. Their organisational structure and key members of the company can also be learnt. By using internet search engines and publication databases, some of the company's publications, representations and patents have been found, from which some of their key technologies, innovations, and new products are learnt.

Document review is also an important method for industrial investigation. Detailed product design as well as information of development processes and activities can be obtained by reviewing design documents, working regulations and communication history. Specifically for this research, a lot of engineering design change cases in their daily work have been reviewed, which is critical to raw data collection for the research. Most of design change cases are stored in their daily working reports and technical reports. A small part of them have been managed in a document management system whilst most of them are still paper-based and stored in their archive room. Some selected design change cases and a lot of design solutions have been reviewed and studied which are used for case study and validation of the methodology proposed in this research.

Focus group interview and unstructured questionnaire are very useful in learning industrial applications and understanding their problems in this early stage of research. Although some information about product design details and technical problems can be obtained from the pre-investigation and document review, a lot of other information is not explicitly stored and largely underlies individual experience. Therefore, face-to-face interviews with key members of the company are necessary in order to get implicit information from people's experience. A questionnaire has been developed for the interviews, which includes five parts, i.e., company and its internal organisational structure, product lines and collaborations, product design and development, engineering change management and knowledge management. Five members who play different roles in the company have been interviewed. They are the chief executive officer, the head of product development, the project manager of the new product, an engineer in the new product team and the manager of the licensing department. Although the interview is based on the same questionnaire with same questions that this research concerns, people from different backgrounds always give different answers from different perspectives or on different levels. That is very important overall to the investigation since relatively more comprehensive information can be collected with different views.

Field observation is also a good way of industrial investigation, through which products and development processes in real world can be learnt rather than just reading from documents or listening to people telling their stories. Most importantly, information obtained from document review and interviews can be materialised by watching engineers working in the field. In this investigation, observations have taken place in the assembly plant which is also the only production unit the company has since they have outsourced most of their production work to local and overseas manufacturing companies. Some design change cases have been reviewed with their effects in the manufacturing stage.

1.6.4 Methodology Development

The methodology development stage is the critical stage for proposing the idea and related techniques to fulfil the objectives of the research. Critical and independent thinking is the major part of work in this stage where the framework of the methodology, the working processes and key techniques are developed. During this development process, specific literature review has also been carried out to find suitable techniques developed by other researchers. The literature selected for reviewing may not be proposed for the domain of this research but they are useful for some technical parts of the methodology. Although these techniques are not a part of the original contributions of the research, they serve as an integral part of the methodology and play an important role in making the methodology workable.

Academic seminars are also used as a method for this research. It is not officially recognised as a research method but found very useful to improve the methodology. During the development of the methodology, objectives of the research are supposed to be fulfilled by proposed methods. It is really a good idea to produce a technical paper or proposal with the stage results at some points during methodology development. Then the technical papers or proposals have been submitted to some academic seminars or conferences. Normally they will be peer reviewed by experts from similar research domains and come back with feedback and comments which are very valuable for improving the methodology. If the work is selected for presentation in a seminar or a conference, it will also be questioned or commented by academic and industrial participants. These questions and comments are also very valuable for methodology development.

1.6.5 Software Tool Development

At the start of the software tool development stage, the general architecture of the software system is designed according to the functionality of the methodology. For each part of the architecture, software tools and special software techniques need to be considered to implement the corresponding functions. A broad review of software techniques, platforms and tools has been carried out. With consideration of the proposed methodology, the techniques and tools are reviewed in terms of their functionalities, usability, platform openness and costs of licensing and using. For those functions of the methodology that cannot be implemented by existing tools, programming work needs to be carried out. This type of work normally involves functional design, coding, testing and integration. Programming tutorials, company white papers, system documentation, and on-line technical communities are very helpful to fulfil the programming tasks.

1.6.6 System Test and Validation

In this stage, the developed methodology and the software tool are examined from potential users' perspectives and therefore the shortcomings are identified for further improvement. The system test and validation carried out in this stage are composed of six steps. In each step, different research methods and techniques are used to fulfil the tasks. In the first step, a structured questionnaire with validation criteria is devised. The questionnaire mainly covers three top level criteria, i.e., functional coverage, tool usefulness and usability. In the second step, potential users are selected who will be asked to carry out the questionnaire and give feedback. In the third step, a case study is carried out with a selected industrial example from the investigated company. In the fourth step, the questionnaire is answered by participants and personal interviews are carried out with potential users via internet-based technologies (i.e., Webinar, video conferencing, email). In the fifth step, feedback and comments from potential users are summarised and classified. In the last step, feedback and comments are analysed and the shortcomings of the methodology and the system are identified.

1.7 Thesis Structure

This thesis is structured in seven chapters which are:

Chapter 1: Introduction. In this chapter, the trends and challenges of the world's manufacturing industry are discussed. New product development and engineering change management are briefly reviewed. The research gaps are identified. The aim and objectives of this research are clarified. The research scope and potential benefit to the industry are stated. The research approach employed in this research is described.

Chapter 2: Literature Review. In this chapter, published research literature with regards to engineering change management, knowledge management in product design and engineering change management and technologies for problem solving are reviewed. The research gaps existing in those proposed methods are identified and discussed.

Chapter 3: Industrial Investigation. In this chapter, a manufacturing company is investigated. Findings from the company regarding engineering design change management and design knowledge reuse are discussed.

Chapter 4: The Proposed Methodology for Engineering Design Change Management. In this chapter, the methodology for engineering design change management is presented. The framework of model-driven and knowledge-based engineering design change management and its related techniques are described in detail.

Chapter 5: System Development. In this chapter, a software system which implements the proposed methodology is described. The architecture of the system is depicted and described. Also key techniques of the system are described in detail.

Chapter 6: Case Study and System Validation. In this chapter, a case study with an industrial example from the investigated company is demonstrated. A questionnaire is developed and potential users of the methodology are interviewed. Their answers and feedback on the methodology and the developed system are summarised and discussed.

Chapter 7: Conclusions and Further Work. In this chapter, the conclusions of this research are stated. Further work of the research is presented.

CHAPTER 2 LITERATURE REVIEW

This research project is dedicated to the development of *a method to analyse engineering change in the product design phase, resolve design conflicts arising from engineering change and evaluate change solutions*. In order to help understand the academic background of the research and justify the academic motivation, a comprehensive review of literature in related areas has been carried out. The purpose of literature review is to learn cutting-edge methodologies and technologies developed by other researchers, to identify research gaps from them, and therefore to justify the aim and objectives of this research. The scope of review of literature covers previous research in engineering change management in the product design phase, and closely related work in knowledge-based methods used in product design and development.

2.1 Introduction to Engineering Change Management

Engineering Change Management (ECM) has been studied by many academic researchers and industrial practitioners in the past decades. Researchers and practitioners investigated various engineering companies and projects in manufacturing industry (Stanev et al., 2008; Tseng et al., 2008a; Huang and Mak, 1999; Huang et al., 2003; Nichols, 1990; Gerwin, 1982; Pikosz and Malmqvist, 1998), construction industry (Chang et al., 2010; Zou and Lee, 2008; Xue et al., 2008; Kraft and Nagl, 2007; Xue et al., 2006), and software industry (Williams and Carver, 2010; Mohan et al., 2008; Park and Bae, 2011; Lam, 1998). Some characteristics of engineering change have been learnt from these investigations and critical issues within engineering change management have also been identified. Therefore, management processes/frameworks, methodologies, technical methods and tools for engineering change management have been proposed in the attempt to address these issues and tackle them with solutions from different perspectives.

2.1.1 Definitions of Engineering Change

What is engineering change exactly? In many cases, engineering change is also referred to with related terms such as engineering change request (ECR) (Helms, 2002), engineering change order (ECO) (Loch and Terwiesch, 1999; Pikosz and Malmqvist, 1998), engineering change notice (ECN) (Buckley, 1996; Ullman, 2010), request for change (RFC) (Keller, 2005) and Change Request (CR) (Lam, 1998; Crnkovic et al., 2003; Kajko-Mattsson, 1999). Although the exact meanings behind these terms may be seen slightly different in terms of the stages of the engineering change process, they represent the same essence of work in companies, and in many times people use them interchangeably. This actually reflects the fact that even if engineering change management has been researched and practiced for many years, the definition of engineering change has not had general consent. Therefore, to consolidate the background of the research topic, it is worthy of having a review of views and definitions of engineering change before rushing into review of methods and techniques proposed in this area.

Wright (1997) defined engineering change as modification to a component of a product before it goes into production. The definition is focused on changes on physical components of products at any stages before production which is mainly in the product design phase in the general product development process (Ulrich and Eppinger, 2004; Pahl et al., 1996).

However, some researchers have defined engineering change as modification to dimensions, fits, forms, functions, materials and so on, to product or components after the product design is released (Huang et al., 2003; Kocar and Akgunduz, 2010). In their view, engineering change has a broader scope where modifications are applied, which includes modifications not just to components but also to the whole product. At the product level, change to function is an important constitute which is one of the significant differences from Wright's definition. Besides, they argued that engineering change occurred after the design of product was released, which included the production phase and all other phases afterwards. Obviously, Wright's definition of engineering change focuses more on the design stage while Huang and Kocar's definition is more about change in the manufacturing phase, which is the essential difference.

In another paper, Tseng, Kao and Huang (2008a) viewed engineering change as modifications to a component or a portion of a product for certain improvement purposes, such as adding more functions, strengthening quality, enhancing aesthetic or operational features, or improving manufacturability. Compared to the previous two definitions, their definition of engineering change is at a higher level and extends the scope that engineering change covers. They also advocated Huang and Kocar's definitions that engineering change should happen after the product design stage or even in the production stage. A notable point made by them was that they thought engineering change was about improvement based on the current design, which included either for adding some values or for reducing costs.

While some other researchers viewed engineering changes as changes occurring in a wider range from customer requirements to product use (Pikosz and Malmqvist, 1998; Eckert et al., 2004). In a joint proposal for Object Management Group (OMG) which is a influential international computer industry consortium, engineering change was defined as ‘a task by which companies request, implement and affect changes to products, documents, components, manufactured or purchased parts, processes, or even supplies’ (Dec et al., 1998). This definition considers engineering change not only within products but also in processes and in the supply chain.

2.1.2 Different Views on Engineering Change

A lot of research carried out in the manufacturing industry reported that engineering change has caused serious problems (Huang and Mak, 1999; Maull et al., 1992; Boznak and Decker, 1993; Kidd and Thompson, 2000). Most of the researchers found that the engineering change process was very time consuming and costly. In Huang and Mak’s survey of 100 manufacturing companies in the UK, they found that there were about 65 active engineering changes on average in each company (Huang and Mak, 1999). In another study, it was shown that companies had 220 change requests per month on average nowadays, 22% of which were manufacturing related changes (Stanev et al., 2008).

The reported figures above are echoed by evidences from other researchers. Boznak and Decker (1993) reported the annual administrative processing cost for engineering change in companies, which they investigated, ranged from US$ 3.4 million to US$ 7.7 million. Maull et al. (1992) found that engineering changes might cost companies about 10% of their annual turnover. Apart from concerns of cost for engineering change, it was also discovered that it might require an average of 40 days to discover an engineering change, 40 days to process and approve an engineering change, and 40 days to implement it (Watts, 1984).

Köhler (2008) cited some German reports about how engineering change negatively affected industry. It showed that engineering change management consumed 30 to 50 %, sometimes even up to 70 % of the capacity in product development. This was supported by Ehrlenspiel (2007) who found that processing changes often absorbed from 20% to over 50% of the product development capacity in the manufacturing industry.

Despite the negative effects of engineering change on product development, some research also showed that engineering change facilitated companies' innovation and creativity. A study by the Aberdeen Group showed that the majority of changes were necessary for innovation (Brown, 2006). There are 82% of interviewed companies stating that although they were mostly concerned about increasing product revenue, engineering changes could also lead to innovation to their existing products. Conrad et al. (2007) also found that one of the reasons for engineering change was the improvement, enhancement or adaptation of a product. It was also supported by Balogun (2003) who saw engineering change as a means to facilitate knowledge generation in industry.

2.1.3 Causes of Engineering Change

By reviewing literature in engineering change management, it is noticed that people's understanding of causes of engineering change have advanced. Research in earlier time was focused on engineering changes emerging from physical components/parts, which included changes of dimensions, materials and forms. Typically, the draft standard ISO11442-6 (1996) gave some examples of engineering change, which included change to a part because of alterations of function or production requirements, change to the application of a part, adding new parts, replacing or removing existing parts, correcting errors in a document or updating a document.

However, Pikosz and Malmqvist saw causes of engineering changes in the view of product development (Pikosz and Malmqvist, 1998), which included changes in customer specifications, faults in the interpretation of customer demands into technical requirements, difficulties in parts fabrication or assembly, weaknesses in the product identified during prototype testing, quality problems with some subsystem or component, and development for future product revisions. Obviously, their view of the causes of engineering change is from a high level of the product development process. Engineering changes are not just seen as modifications to physical parts but to possible changes in any stages of the product development process.

Dennis et al. (2002) viewed engineering change from a system engineering perspective. They identified that engineering change might be caused by (i) problem reports that identify bugs that must be fixed, which forms the most common source; (ii) system enhancement requests from users; (iii) events in the development of other systems; (iv) changes in underlying structure and or standards (e.g., in software development, this could be a new operating system), and (v) demands from senior management.

Eckert et al. (2004) saw engineering change in two main categories: initiated changes and emergent changes. Initiated changes are those intended by a stakeholder, while emergent change is unintended and occurs when some aspect of the system design requires changing because of errors or undesirable emerging system properties, often due to an earlier initiated change.

Conrad et al. (2007) saw the causes to engineering change at a higher level. They stated that engineering change could be triggered by (i) a change of one or more characteristics, e.g., a modification of the diameter of a shaft; (ii) a change of one or more required properties, e.g., the change of customer desires; (iii) a change of external conditions, e.g., new standards; (iv) a change of internal dependencies, e.g., the realisation of a different solution; (v) a change of relations between characteristics and properties, e.g., the use of a different formula, tool or practical experiences from the field.

While people’s views of causes of engineering change have been changing over time, the focus and the scope of research on engineering change has been widened. The initial motivation of studying ECM was to avoid engineering changes during the manufacturing process due to the adverse effects they caused. The adverse effects caused in terms of delivery time, developing cost and product quality are noticeable, but very difficult to estimate (Huang et al., 2003). Later on, people realised that engineering changes were actually inevitable. Therefore, researchers have turned to finding out how engineering changes went on and what kind of impacts they might cause (Clarkson et al., 2004; Kocar and Akgunduz, 2010; Ouertani, 2008). Recently, some researchers argued the benefit of engineering changes to innovation and creativity, which could enhance the competitiveness of companies. Thus some researchers have started to study engineering changes from perspectives of knowledge management and knowledge reuse (Balogun and Jenkins, 2003; Palani Rajan et al., 2005; Lee et al., 2006; Keese et al., 2009).

2.2 Change Propagation Analysis

One of the most difficult issues in engineering change management is that when a component is changed, it may also change its related components (Ariyo et al., 2009). Therefore, an initial change may cause changes spreading at several structural levels and across different functional domains. A formal definition of change propagation was given by researchers. They considered change propagation as the process by which “change to one part or element of an existing system configuration or design results in one or more additional changes to the system, when those changes would not have otherwise been required” (Giffin et al., 2009; Eckert et al., 2004). Essentially, the reason why changes propagate is the dependency between components (Eckert et al., 2004). This situation is so-called change propagation or the knock-on effects of changes, which makes change analysis very tricky. Methodologies and techniques regarding change propagation analysis are reviewed in this section to see what researchers have done to tackle this problem and the shortcomings of their methods which can be possibly improved by further work.

Giffin et al. (2009) summarised the recent research interests in change propagation:

- Descriptions of the nature of change propagation, which state the reasons for interest in the field,

- Methods for controlling change propagation through better design decisions. This includes work in “design for changeability”, which emphasizes designing products and systems in the first place, and

- Development of tools for predicting change and visualizing networks of changes

2.2.1 Analytical Methods for Change Propagation Analysis

Clarkson et al. proposed a method called change prediction matrix (CMP) to trace change propagations and analyse the impacts they may cause (Clarkson et al., 2004). The method transformed the dependency relationship between components in a product model into a design structure matrix (DSM). Based on this matrix, the likelihood that potential change propagations might happen between components were estimated. Also in the same way, the impacts of these potential change propagations were estimated. By combining the change likelihood matrix and the change impact matrix, a change risk matrix was generated. With help of visualising method, change propagation paths and their relative risks were clarified.

In another research, Eckert et al. (2004) proposed a method to analyse change propagation at a parametric level and identify four types of change propagation behaviours, namely constants, absorbers, carriers, and multipliers. These four types of change propagation behaviours helped to analyse change propagations that cross multi-levels. Four types of change propagation behaviours represented four situations when a change of a component propagating to another component via some other components. They included changes being passed without effect, being reduced or eliminated, being replaced with new changes from the intermediate component, and being enhanced. This method was also integrated with the CMP method to enhance the performance of change propagation analysis in product conceptual design (Keller et al., 2009).

Suh et al. proposed a method, called change propagation index (CPI), to measure the degree of change propagation caused by an element in the system when there was an engineering change imposed (Suh et al., 2007). They adopted Eckert's definition of change behaviours that an element performed in response to engineering change. As reviewed above, the change reactions are defined as multiplier, carrier, absorber and constant (Eckert et al., 2004). The degree of change propagation (the CPI) of an element is calculated in a matrix by subtracting the number of elements that are changed by the element from the number of elements causing changes on the element. If the result is a positive value, it means the element is a multiplier. A negative value means it is an absorber. A zero means it is a carrier. If there is no element causing change on the element, nor changed by the element, that means it is a constant. They also estimated change cost of each element in the system (K_{switch}). By combining the CPI and K_{switch}, the whole impact of a change propagation route can be estimated in terms of the change costs. Although this method qualitatively formalises Eckert's definition of change behaviours, it is far from accurate. The impact of change propagation from one element to another cannot be simply estimated by a binary value (0 or 1) since the impact depends on the essence of the interaction or connection between them. Also, the change impact largely depends on the extent to which the causing element is going to change.

Researchers in MIT conducted an investigation in engineering change management with 41500 change requests (Giffin et al., 2009). They claimed that it was the biggest data set that had been investigated among the current published product design literature. They used the change prediction matrix (CPM), a method proposed by Clarkson et al. (2004), to analyse the large set of engineering change requests. One interesting finding from this analysis is that there are a lot of change propagations where direct structural connections do not seem to exist. This shows some relationships between components may not be identified in the CMP method but they actually cause change propagation.

They also used graph theory to analyse the data set and tried to find engineering change patterns. They found that most of the components were related to fewer than 10 changes while a small number of components related to a huge number of changes. They also recognised that the engineering change network was very complex and basic change patterns might be useful for clarifying the relations between engineering changes. They proposed three types of change patterns (motifs), namely, the '1-Motifs' pattern representing a single and isolated engineering change, the '2-Motifs' pattern representing the parent-child type change network or the sibling type change network, and the '3-Motifs' pattern representing a hybrid change network of the parent-child type change and the sibling type change. However, only the parent-child change network was used to analyse change propagation. The other patterns were either too simplified or too complex to use for analysis.

Following on from the work of Suh et al. (2007), they enhanced the algorithm to calculate the change propagation index (CPI). The CPI was calculated by dividing the difference between the incoming changes and the outgoing changes to an element by the sum of both of them. Therefore, it came out with a value ranging from -1 to 1. The value could be used to indicate whether it is a multiplier, an absorber, a carrier or a constant (Eckert et al., 2004). Although this improved method still cannot solve the problem with Suh's method which, as mentioned above, is difficult to analyse and predict change propagation during product development, statistically it can help engineers identify which part of the system is more likely to have change propagation happening based on the large sample of engineering change cases.

Koh and Clarkson (2009) proposed a modelling method for engineers to track change propagation during product design. Rather than just focusing on change propagation in physical components as in some other similar methods, their method considered change propagation analysis as a way to pre-examine the design plan and improve resource allocation before going to embodiment design. This method employed a multiple domain matrix to model and to qualify interdependent and interactional relationships between product attributes, design features and components. By using the change prediction matrix (Clarkson et al., 2004), change propagations between components were identified and their impacts on related design features were quantified. Then an algorithm was developed to revise the performance ratings on product attributes with consideration of change impacts on related design features. As a result, a performance rating chart was populated to compare performance ratings on product attributes with and without change propagation analysis. The method defined a clear and intuitive way for engineers to analyse design change propagation and to adjust their expectations of performance of product attributes, which could benefit optimisation of design plan and resource allocation. However, there are also some difficulties in using the method. Predefinitions of relationships between product attributes, design features and components are roughly estimated so that the accuracy of performance rating calculation is questionable. Also according to responses from engineers, the method is not easy to use since the modelling approach is too complex and not user-friendly.

2.2.2 Tools and Techniques for Change Propagation Analysis

Kocar and Akgunduz (2010) proposed a different method to analyse change propagations. They used visualisation technique and data mining technique to represent product models and find out dependencies between components. Users would be warned visually if potential change propagation was predicted to happen.

Ouertani and Gzara (2008) proposed a visualisation tool called DEPNET to track change propagation by modelling product specification using UML technology and oriented graph throughout the product design process. Differing from matrix-based methods for change propagation analysis which are focused more on relationships between physical components, this method is focused on dependency between product specification which has a broader scope including functional, structural, behavioural and geometrical relationships. Four types of dependency relationships were discussed including redundancy relationship, consistency relationship, dependency at creation and dependency at modification. Based on qualitatively estimation of these relationships, an equation was proposed to calculate the degree of dependency. The dependency network composed of nodes (product specifications), directed arcs (dependencies) and value of arcs (dependency degrees) was built up for change tracking and impact estimation. However, the elements used to form dependency degree calculation were purely based on engineers' estimations on each individual design case. It is difficult to always have consentaneous estimations from case to case, which may cause the method to be imprecise.

Ma et al. (2008) proposed an algorithm for engineering change analysis based on a product modelling method which was called unified feature modelling scheme. The modelling method was proposed by them to clarify the relations between elements of product design by using the unified feature definition and the data association mechanism (Chen et al., 2004). They used this modelling method to clarify the dependency relationships within product design, which was intended to facilitate change propagation analysis. The dependency relationships identified in this modelling method include intra-stage geometric and non-geometric constraint relations which are direct dependency relationships within a stage, and the inter-stage data associations which are indirect dependency relationships across different stages. Although by using the proposed modelling method they managed to clarify important dependency relationships within product design, the algorithm for identifying change propagation is limited. Their algorithm starts to check the constraints between the to-be-changed variable and its related variables from the intra-stage. If the change within the stage is acceptable, then it moves to check associated features or properties outside of the stage (inter-stage). Every related feature or property is then checked as what is done in the intra-stage. A major problem is that, in the intra-stage, if a related variable is a driving variable or it's a driven variable but with no alternative values available to satisfy the constraint which is initially violated by the to-be-changed variable, then the change request is simply rejected. In this way, the change propagation process is not iterative and very limited. In reality, an engineering change may cause change on its driving variables (upstream variables) and even on predefined constraints. It all depends on whether the impact that change may cause is acceptable and viable. However, this method does not provide a way to evaluate the impact of a change.

Do et al. (2008) proposed a product data based method to track engineering change propagation and therefore keep the product data consistent throughout the collaborative product development process. To avoid inconsistency while making changes on a product in different functional teams, the proposed product data model has a base product definition with different product views which are correspondingly for different functional teams. Changes made in different views are actually stored in the same base product definition model. Thus, changes made in any product view can be consistently presented in other product views. The method helps maintain information consistency of the product model while changes happen, which is important in a collaborative product development environment. For example, in such an environment, changes in the design stage can be effectively passed to the manufacturing stage by using this method, which is critical for change propagation since downstream changes largely depend on the previous change. The weak link of this research is that the proposed method uses a lot of historic data of engineering change to help maintain the consistency, which may cause a certain delay of corresponding change to the base product definition and its product views. Integration of the proposed method with the engineering change process may be more beneficial.

2.3 Impact Analysis in Engineering Change Management

Impact analysis is an important task for engineering change management. A lot of research work has been done in this topic (Mckay et al., 2003; Ouertani, 2008; Ouertani and Gzara, 2008; Xue et al., 2006; Xue et al., 2008; Zou and Lee, 2008; Lee et al., 2010; Rios et al., 2007). In this section, recent publications in change impact analysis are reviewed.

Mckay et al. (2003) proposed a methodology called "Feature Elasticity" to evaluate change impact in the early stage of product development in an extended enterprise. They claimed that the methodology could determine the impact of change on a product's design based on its effect on the relevant process plan which was a part of a distributed process planning system. In their work, the authors considered four metrics to evaluate the overall performance of each process plan, which were quality, cost, delivery and knowledge. An objective function was also proposed to convert impacts on all four metrics to equivalent values in cost, so that trade-off can be made to achieve the best performance. They identified two types of changes on product features, namely elastic change and plastic change. An elastic change was considered in their research as one which did not change the production routes so that selected processes and resources were still valid after the change. They also considered that elastic change had a "spring-like" impact on those four metrics of process planning performance. Therefore, when a change of product feature happens, the objective function can be used to evaluate the final change impact and help designer make decisions. The major advantage of this method is that change impact on each metric can be estimated so that by using the objective function some trade-off decisions can be made in order to get an optimal result. However, this methodology has not considered the effect of change propagation which may cause other knock-on change effects on the product model and the process plan. Therefore the "spring-like" change impact assumption may become invalid, which may cause the results to be imprecise.

Tseng et al. (2008a) also proposed a methodology for evaluating engineering change in a collaborative product development environment across multiple plants. However, their approach is focused more on design change and its impact on collaborative manufacturing operation. They also proposed an objective function to evaluate the change impact on manufacturing operation across multiple plants. The objective function is formalised by a comprehensive set of factors that affect the total cost of manufacturing operations, which include manufacturing cost, transportation cost, material cost, labour cost, machine operation cost, and assembly operation cost. By using the objective function, the change cost of manufacturing operation, which is caused by component changes, can be optimised via properly distributed changes in collaborative plants. After evaluating all of the alternative design cases, the one having best value can be chosen by comparing the original cost of manufacturing operation before the design change happens. The idea of their methodology is similar to Mckay's (2003). However, their method has a more comprehensive consideration of impact factors on manufacturing operation. But also their method does not consider the change propagation thoroughly before evaluating the change impact. Knock-on changes caused by the initial change may significantly change the results from their current method.

In another paper, Tseng et al. (2008b) proposed a different method to evaluate engineering change impact in terms of the relationships between components (or say degrees of dependency) based on five relational factors. They assumed that for each change request there were various candidate engineering change cases. In each engineering change case, some of the components are to be changed and then some related components may be potentially affected. The relationships between to-be-changed components and potentially affected components are qualitatively evaluated in five different matrices according to five relational factors. The relational factors include dimension, plant location, material, manufacturing operation and assembly operation, which are considered as important to collaborative product development.

Therefore, for each engineering change case, there are five summed relational values, which are generated from those five evaluation matrices corresponding to five relational factors. After that, the AHP (Analytic Hierarchy Process) method (Saaty, 1988) is used to compare and weight the relational values. The weighted relational values are then used to calculate the final relational value for each engineering change case. Engineers can then select most promising solutions for the engineering change by prioritising final relational values of candidate design cases. An unanswered question in this method is that: how can the related components be determined? The authors have not given an approach to identifying related components for analysis. Also they have not clarified what metrics they used to judge an engineering change case with smaller relational value were better than others.

Lee et al. (2010) proposed a method to evaluate engineering change impact in modular product development using the ANP method (Analytic Network Process). The authors claimed that previous research in engineering change impact was focused only at part or component level in integrated product architecture. There was a lack of support for change impact evaluation for modular product development. In a modular product, an engineering change may not only affect elements within a module, its impact may also be spread to elements across multiple modules via interactions between them. The ANP method, proposed by Saaty (1996), is generalised from the AHP method for multiple criteria decision making. By using ANP, the method was claimed to be able to capture and weight direct and indirect change impacts across different modular levels. Therefore, each part of the product is assigned with a value of relative change impact (RCI) which is generated from the ANP approach. The authors actually proposed a method using ANP to analyse the interdependency between elements in modules. The novel point of this method is that it enables engineers to analyse interdependency between elements from different modules and provide a qualitative way to evaluate the degree of interdependency of each element. However, since the method is based on predefined and static product model, the dynamic characteristic of change propagation is not captured. The dynamic characteristic of change propagation means that the impact of a change largely depends on the nature of the change itself and the relationships between parts. In many cases, although there is more than one type of relationships between two parts, a certain change of one part may just impact the other one through one certain type of relationship (e.g. spatial or behavioural). Therefore, it is not appropriate to evaluate change impact using a synthesised degree of dependency which is considered with all relationships between involved parts.

Conrad et al. (2007) proposed a FMEA-like method, called change impact and risk analysis (CIRA), to evaluate the impact and risk of engineering change. They developed a form which was similar to the FMEA form. In this form, the design element which requests change and its relevant design elements are identified. Also characteristics of the design that can be changed to satisfy the change request are listed and corresponding solutions are proposed. Based on those proposed solutions, other design elements and characteristics that may be affected are identified as well. In the end, qualitative methods are used to assess three metrics including the significance of affected design elements, likelihood of success of the change solutions, and the degree to change impacts. Qualitative values of these metrics are then multiplied to generate a value called change classification number (CCN) which is supposed to help engineers make change decisions. Change solutions with lower values of CCN are considered as better solutions. Their method provides an intuitive way to estimate impact of engineering change. However, a systematic approach is lacking to qualitatively estimate those three metrics. The proposed metrics are at a very high level of abstraction so that simple and overall judgements for their values may not be precise enough.

Steffens et al. (2007) viewed change impacts from the perspective of project management in complex product development. Based on 7 projects that they investigated, they found there were five types of criteria that were used by the project team or the product-line management board for making change decisions in product scopes, which included project efficiency, customer impact, project portfolio, business success and preparing for the future. They also found that decision making shifted between the project team and product-line management board from project to project. Although the work is more about investigating criteria for change decision making, the criteria they proposed are actually very useful while considering the engineering change impact in terms of project success.

2.4 Computer Aided Tools for Engineering Change Management

Although methods for engineering change management have been proposed with rigorous analytical or reasoning approaches, designers will be easily exhausted before having investigated the entire search space (Ariyo et al., 2009). Therefore, it is important that computer aided tools are developed and used in engineering change management. Previous investigations have shown that computer-aided ECM systems have not been regularly used in companies (Pikosz and Malmqvist, 1998; Huang et al., 2003). Some companies have used electronic document management systems to replace paper-based ECM documents, but the data are not structured and they cannot be systematically managed.

Although currently not many companies are using structured computer-aided systems to facilitate their engineering change management, there are some systems that have been developed by academia to enhance communication and information sharing in change management process. Huang et al. (Huang et al., 2001) developed a web-based system to implement the whole process of engineering change management, including engineering change log, engineering change request, engineering change evaluation and engineering change notice. The distributed system improved the efficiency of ECM and enhanced the collaborations between engineers. Also structural ECM data made it possible to integrate with other computer-aided systems such as PDM, CAPP, CAD.

Chen et al. (2002) identified a critical task in engineering change management was to 'ensure that the latest version of modifications, product and process data items and specifications are in the right place at the right time.' This became an important issue in allied concurrent engineering. Therefore, they proposed a methodology from the viewpoint of enterprise integration to integrate engineering change management processes, project management, and product information. Three types of integration were used including process integration, information integration and system integration. An engineering change management framework was proposed to associate the project structures between engineering change management and allied concurrent engineering. Following on, the system model for engineering change management in allied concurrent engineering was proposed. A system based on the client-server architecture was developed which integrated the engineering change management process with allied enterprises for concurrent engineering projects.

Ouertani and Gzara (2008) developed a system called DEPNET to visually track dependencies within product specifications so that change propagations could be captured if any design changes of a product specification happened.

Kocar and Akgunduz (2010) developed a visualisation system to track change propagation which was well integrated with a 3D modelling system. The system is able to predict potential engineering changes which are caused by other changes, and prioritise by using a data mining approach.

Lee et al. (2006) developed a knowledge based system to facilitate engineering change management in a collaborative environment. The authors used ontology technology and case based reasoning method to construct a knowledge base of previous development experience. It also implemented the knowledge base with a web-based system which enabled users go through the whole engineering change management process from change request initiated to change approved.

2.5 Knowledge Based Methods for Product Development

Knowledge is considered as the key asset of enterprises that contributes to enterprise competitiveness and provides the basis for long term growth (Kalpic and Bernus, 2002). Currently, new product development seriously depends on individual experience, which makes the development unstable and uncontrollable. It is realised that knowledge obtained from previous projects is hosted by individual experts, and in turn, individual experts use previous knowledge into new product development and obtain new knowledge from it. This makes knowledge share and reuse in development team inefficient and furthermore makes the project risky. In order to facilitate knowledge acquisition, knowledge storing and knowledge sharing to be more efficient, research has been carried out by many researchers to use computational knowledge management technology in new product development.

Knowledge based engineering (KBE) is widely recognised as an efficient way for knowledge management in engineering area. Knowledge-based engineering method is designed to capture information of product and process in such a way that allows businesses to model engineering design processes and then automate all or part of the process, which facilitates to acquire, represent, store, reason and communicate with the intent of design processes (Chapman and Pinfold, 2001). Lovett et al. (2000) considered KBE as a method that captured expertise of experts in specific application domains, incorporated it and made it available in computerised systems.

Best design practices and engineering expertise are captured and formulised in a shared knowledge base which can contribute to following product development. There are many cases that use KBE methodology to help product development, which are proven to be successful. Textron Aerostructure stated that 73% of design time was reduced by developing a knowledge based tolling design application (Brewer, 1996). Jaguar was reported to manage to reduce the design time of an inner bonnet from eight weeks to 20 minutes by using a knowledge based engineering system (Kochan, 1999). Also Boeing announced that approximately 20 000 parts of the 777 had been designed using KBE (Heinz, 1996). The US Air Force Research Laboratory published papers to study the development of a structural modelling tool using KBE techniques, which was supposed to address structural concept designs of an uninhabited combat air vehicle. British Aerospace (BAE) used KBE in the development of A340-600, which was proven to get a huge reduction of developing time (Oldham et al., 1998).

2.5.1 Ontological Methods for Product Development

Ontology based methods for engineering product design have attracted many researchers' attentions in recent years. The term 'ontology' originated from philosophy to describe the nature of being (Alter, 2009). In the computing and engineering world, ontology is more commonly seen as 'a formal, explicit specification of a shared conceptualisation' (Studer et al., 1998). Therefore, a lot of engineering research used ontology as a measure to facilitate a shared understanding of a knowledge domain that might be communicated among people or software agents (Gruber, 1993; Pinto and Martins, 2004; Corcho et al., 2003). In the remaining part of this section, a review of relevant research work is carried out to show what researchers have done to apply ontological knowledge-based methods in product development.

Kitamura et al. have published some papers in recent years that used ontology-based method to formalise functional representation of product design (Kitamura et al., 2004; Kitamura et al., 2006; Kitamura and Mizoguchi, 2003; Kitamura and Mizoguchi, 2004; Kitamura and Mizoguchi, 1998). They found that it was difficult in the industry to represent the functionality and components of a product. An ontology-based functional representation approach was proposed by them to address this issue. The approach is based on a functional representation technique and ontological definitions of functions and physical devices. The functional representation technique was used to decompose a product in the functional domain, the physical domains and the relationships between them. Then the predefined functional ontology and physical component ontology were used to formalise elements of the decomposed product model. An example was used to demonstrate the method used in a mechatronic product. A process and instruction of using this method were described during the example.

Chang et al. (2008) developed a method for product conceptual design, which was based on ontological technology and IDEF0 modelling method. In their approach, a modified IDEF0 modelling method is used to capture design knowledge from product development. A consistent ontology definition is also designed to define elements of conceptual design, i.e. functions, flows, and components. Therefore, design knowledge captured from modified IDEF0 models can be transformed into an ontological knowledge base. An algorithm for similarity comparison and a query method based on relational database were also proposed. By using this method, designers can input new functional requirements and find semantically similar design case.

Brandt et al. (2008) proposed an ontological method for knowledge management in the product design process. The method is built based on a core ontology which covers conceptual definitions in four domains, i.e. product, process, description, and storage. A computer aided system was also developed to formalise the knowledge base with ontology and relational database. Knowledge from product design, development process, management and other technical documents were transformed by using the proposed ontology and stored in the database. A guideline of using the knowledge base to retrieve and reuse design knowledge was also provided.

Bradfield and Gao (2007) found there were knowledge sharing barriers in new product development processes in multinational company where designers might use multilingual working languages. They used protégé, an ontology editor, to build ontology to capture and formalise the process and task knowledge in new product development. The ontology was designed to be able to be accessed and reasoned in different languages by using attached multilingual metadata. Therefore, process and task knowledge captured from a company can be reused in another company in different countries using different languages.

Hirtz et al. (2002) developed a fundamental ontology definition for engineering design, by reconciling and improving previous related research efforts, in a proposal for the US department of commerce. In this proposal, the ontology covered taxonomic definitions of flows and functions. They explored other research efforts and enumerated as much terms as possible which defined the flow types and functional behaviours.

Darlington and Culley (2008) developed three types of ontology for supporting to capture engineering design requirement, which included the engineering design requirement ontology, the product finish ontology and the machine motion ontology. The engineering design requirement ontology defines the concepts, data entities and relationships existing in engineering design requirements. The purpose of the Product Finish Ontology is to identify the conceptual elements and their labels related to the specification of the component and product surface finish. The machine motion ontology, which is similar to the functional basis ontology which is defined by Hirtz et al. (2002), defined the functional concepts in product conceptual design. Each ontology definition developed in their paper was evaluated. Software systems that were developed to use those ontology definitions are also discussed.

2.5.2 Case-based Reasoning Methods

There is a saying in the design world that ‘all design is redesign’, which means that most design work is based on existing designs and is actually incremental design. This saying is reflected in a lot of research work. For example, Dieter and Schmidt (2009) defined five types of engineering design in their book ‘Engineering Design’, namely original design, adaptive design, redesign, selection design and industrial design. Excluding industrial design, the first four are technical design and the last three of them are actually incremental design which takes an overwhelming percentage of engineers’ everyday work.

The Case-Based Reasoning (CBR) method was invented to reuse previous cases to solving current problems, which was taken as an advantage and used in many industrial and academic applications. This advantage actually matches the nature of most of engineering design activities. A lot of research has been carried out to use the CBR method in engineering product design.

CBR as a useful problem solving technique by reusing or adapting previous cases has been researched for many years from the 1980s. The basic idea of CBR was first introduced by Schank and Abelson from Yale University in their book investigating human knowledge structure (Schank and Abelson, 1977). However, it was recognised that the first computer based CBR system was developed by Kolodner (1983a; 1983b; 1993) who implemented the dynamic memory model from Schank’s work for a travelling system. While in the design area, the CYCLOPS system which was developed by Navinchandra (1991) was recognised as the first CBR system for interactive design (Goel and Craw, 2005). The CYCLOPS system however did not fully implement CBR functions. It worked mainly as storage system with specific annotations for design case finding. Then Goel et al. developed another CRB system called KRITIK which was recognised as the first autonomous CBR system for design that fully implemented tasks needed, including design case retrieval, adaption, evaluation and storage (Goel et al., 1997). Since CBR has been invented and implemented, it has been used in a lot of areas both in industry and academia.

The general case-based reasoning method is normally composed of five steps (see Figure 2-1) which are briefly described as below: (Aamodt and Plaza, 1994; Kim and Han, 2001; Kolodner, 1993)

- Index assignment. All prior cases need to be indexed by some mechanisms to characterise them. Therefore, users can find proper cases by following the indices.

- Case retrieval. The indexing mechanism is used to find similar prior cases from the case base by comparing the similarities between the problem and prior cases.

- Case reuse. The retrieved cases are selected according to certain measures of specific applications and used to solve the problem.

- Case revision. If the retrieved cases cannot meet the requirements of the problem, proper revision is carried out to create a new solution based on the retrieved cases.

- Case retention. The revised case needs to store in the case base as a new case which is assigned with a proper index.

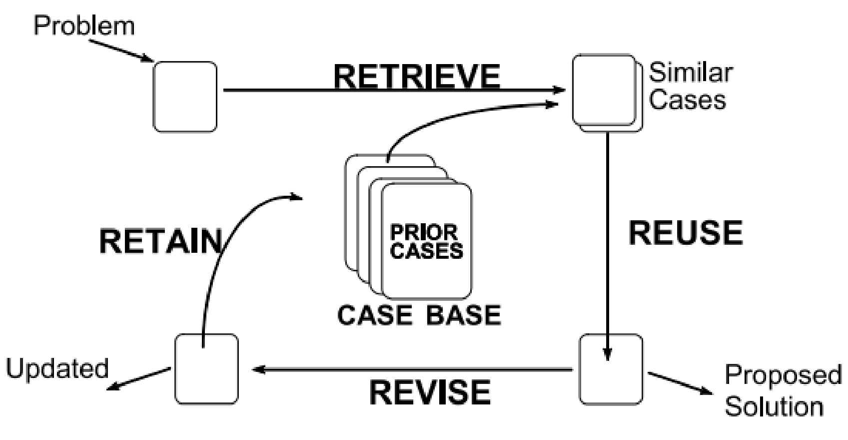


Figure 2-1 The CBR process adapted from (Aamodt and Plaza, 1994)

Suh et al. (1998) developed a case-based expert system for quality design by reusing previous design cases. Their method is generally composed of three steps: (i) Identify quality properties; (ii) retrieve similar design cases using a similarity comparison method; (iii) modify the most similar design case to get a new design. They proposed a hierarchical structured model to represent design cases which included design objects, attributes, and design parameters. Based on this model, they also proposed a similarity comparison method to retrieve structured design cases. This method compares formalised target attributes, which is derived from customer requirements, of a new order with formalised target attributes and result attributes of existing design cases. An algorithm was developed to get quantitative similarity values. Therefore most similar design cases can be retrieved for further modification for the new design. This approach is more effective than other text-based indexing method since it used structured design parameters as indices of existing design cases. Therefore, the similarity comparison algorithm can capture detailed design parameters and compare them with quality characteristics of the target design. However, because of the same reason, the method is not suitable for non-parametric design.

Wu et al. (2008) proposed a case-based reasoning approach to helping conceptual design which was integrated with fuzzy AHP (Analytical Hierarchy Process) method. They employed a “retrieving-and-filtering” process to find design cases and screen them according to design thresholds of the baseline product. The method is innovative in terms of the way they indexed existing design cases. Rather than using design attributes as indices, they used use-scenarios of the design as indices. They proposed a list of use-scenario attributes and also proposed a weighing method to evaluate use scenarios of design cases using fuzzy AHP. Also an algorithmic similarity comparison method and a screening method were developed. The approach is considered having better performance than “retrieve-only” methods but the overall performance for conceptual design has not been fully proven.

Mok et al. (2008) proposed a hybrid case-based reasoning method for injection mould design. The case-based reasoning method used in their approach is integrated with a general knowledge base for mould design. They discussed the advantages of the CBR and general knowledge base system (KBS) and used them to help the design of moulds. In this approach, CBR provides a measure to retrieve previous design cases and then the KBS is used to adapt the retrieved design case and evaluate the new design. The indexing mechanism of the CBR is composed of information about the part, the mould and relationship between them. General design feature of the part and the mould have been enumerated and encoded. This approach takes the advantages of the CBR method and the KBS method and is incorporated with them to help mould design. The indexing mechanism is specific and can be easily implemented in computer system. However, the general knowledge base is design document based. Designers need to look up the knowledge base to find the knowledge they need while doing adaption and evaluation, which is not effective or efficient.

Han and Lee (2006) used case-based reasoning method to reuse previous conceptual design cases in conceptual synthesis of mechanisms. They analysed the primitive mechanisms which served as most basic unit of mechanisms. All other mechanisms are composed of one or more primitive mechanisms. This theory is used to formalise conceptual design cases of mechanisms. These mechanisms are represented by primitive mechanisms in some certain sequences, which are also considered as the functions of mechanisms and used as the indices. A synthesis method was proposed as a reasoning approach to retrieving conceptual mechanism design cases from the case base. The method is actually a combination of case-based reasoning and model-based reasoning. The way of modelling mechanisms using primitive mechanisms is novel and effective, by which a complex mechanism can be clearly structured and abstracted. However, the reasoning process is not very efficient and computational expensive. There is further work needed to do to effectively computerise the methodology due to the complexity and inefficiency of retrieving design cases.

Janthong, Brissaud and Butdee (2010) proposed a product design methodology for mechatronics products. The methodology combines the Axiomatic design theory (Suh, 2001) and the CBR method to retrieve previous design cases. The indexing mechanism developed in this methodology is based on the axiomatic design theory which decomposes a design case into two domains, i.e. the functional requirement (FR) domain and the design parameter (DP) domain. Design parameters in the axiomatic design theory actually mean solutions for the corresponding FRs. There is a mapping relationship created between FRs and their corresponding DPs, which not only links the FRs with the DPs but also links their attributes together. Therefore, when a new design problem comes in, it will be decomposed into FRs and related attributes. The decomposed model of the FRs will be compared with formalised design cases in the case base. An algorithm for similarity calculation was developed to retrieve the most similar design case as reference design. Then if the FRs and attributes of the retrieved design case do not exactly match the design problem, general knowledge will be used to adapt the reference design case to get a new solution. This is a novel method to use axiomatic design and CBR to facilitate innovation of product design. The advantage of this method is that the effectiveness of retrieving existing design cases is improved by using the axiomatic design approach to decomposing and generalising them. Retrieved design cases may be more functionally useful. However, the difficulties of using this method are also notable. Firstly, it needs skilled designers to decompose and formalise design cases using the axiomatic design method. For complex products, this will be extremely time-consuming. And also, there is a lack of measure to keep the consistency of terms used in design case formalisation, which may have significant impact on automated similarity calculation.

Takai (2009) used case-based reasoning method to estimate development cost of a product concept. In his method, existing products and their related attributes regarding costs, functions, technologies, requirements and specifications are stored in a case base. These attributes serve as indices of product cases. Then a hierarchical clustering method is used to retrieve most similar product cases. The similarity, called homogeneity in the paper, is assessed by a binary distance matrix. The matrix is conducted to get the Euclidean distances between concept and each product cases, which shows the relative similarities. By using the clustering methods, this approach can retrieve very similar design cases but the restriction is that the sample space needs to reach a certain scale so that the results can be trusted. Thus, this method may be very useful for general products or in a big organisation producing a lot of similar products. But the usability may be limited in SMEs where product design information may not be sufficient enough for this method.

2.6 Summary

Based on the literature reviewed above, three main research gaps in engineering change management have been identified.

Firstly, current methods do not take into consideration of functional requirements in engineering change analysis. In the design phase of product development, besides changes among physical components, it is also important to consider the effect that physical changes may cause to functional requirements. Thus, changes to one component may potentially damage the realisations of other functional requirements, which may not be identified without considering the relationship between the functional domain and physical structure domain.

Secondly, most research on change propagation analysis uses an estimating and predicting approach to evaluate and predict the knock-on effects of engineering change. It is realised that if a change (the initial change) to one component causes propagated changes to others, the change effect largely depends on the solution of the initial change. That means if there are alternative solutions for the initial change, each candidate solution may cause different impacts on the other components. Therefore, each different candidate solution may cause different propagated changes. For this reason, a prediction method for change propagation, which depends on predefined dependency relationships, may not be effective since the dependency relationships between components are change-dependent and cannot be predefined.

Thirdly, there is a lack of method to help engineering designers reuse knowledge from previous design cases to solve design conflicts emerging from change analysis. Although a lot of research work has been carried out to use knowledge-based method to facilitate product design and development, few of them have tried to use knowledge-based method in engineering change analysis. It is also found that case-based reasoning is an effective method for problem solving which has a great potential to help solve design conflicts in change analysis.

In this thesis, a model-driven methodology has been proposed to consider both functional requirements and physical components in engineering design change analysis. Based on the relationships in and between functional requirements and physical components, a causal approach has been proposed to trace and analyse change propagation and to find design conflicts. Also, previous design case has been formalised and reused as design knowledge to help resolve design conflicts, find and evaluate design change solutions.

CHAPTER 3 INDUSTRIAL PROBLEM AND REQUIREMENT INVESTIGATION

Managing various changes in new product development is one of the most important challenges of engineering research, and the outcome of this research is supposed to benefit the manufacturing industry by helping engineers solve problems they have in their everyday work. Therefore an industrial investigation has been carried out to understand problems or limitations that companies have in the area related to the research topic. During this research project, Vensys Energy AG, a wind turbine design company in the green energy sector, has been primarily investigated along with some other manufacturing companies, in order to learn processes and methods they employ in engineering change management, and more importantly to identify challenges they face.

In this chapter, the company, which has been primarily investigated, and its new product are introduced. The reasons that this company is selected for investigation are discussed as well. Investigation methods used in this research are also described. Some important findings from the investigation are presented.

3.1 Introduction

Industrial investigation is one of the most important parts of research in engineering, since the results of the research are supposed to be helpful to solve engineering problems. Therefore, organisations and their products and/or processes selected for investigation should match the nature of the research topic. In terms of engineering research, it would be difficult to investigate an organisation doing business in a very different area from the area of the research topic. As for this research, there are three basic requirements which need to be satisfied when selecting an organisation suitable for industrial investigation: (i) the organisation is in the manufacturing industry in order to match the expertise and interest of the research group; (ii) product design plays an important part in the organisation since the research is focused on the engineering product design stage; and (iii) the engineering products that the organisation develops are relatively complex so that change management would play an important role in their product design and development activities.

Additionally, it will be more valuable if the organisation is in a promising industry and can typically represent many other companies in terms of its pioneering position in the industry. Although it is not essentially different between investigations of companies in a traditional industry and in a promising industry, there are more opportunities for this research to find new problems in a promising industry and therefore make greater and more valuable contribution, since the promising industry represents the future and also have lots of uncertainties. Vensys Energy AG, which is a Germany-based wind turbine design company, is considered as an ideal organisation for this research.

3.2 The Investigated Company and its Products

The company selected for investigation in this research is a research and development oriented manufacturing company in the green energy industry. Vensys Energy AG is a Small and Medium Enterprise (SME) focused in the area of gearless wind turbine design and development. They design gearless wind turbines and sell production and deployment licenses to wind energy companies. It also has strong international and distributed collaborations. They have worked very closely with local and international manufacturers and wind energy companies to produce and install wind turbines across the world. There are nearly 2000 wind turbines based on their solutions deployed in Europe, Asia and America as of May 2010. They have two product lines at the moment. One is the old 1.5 MWs (million watts) wind turbine which has been in production status for a couple of years and still in continuous improvement. Most wind turbines that they have installed so far are based on this solution. The other one is the enhanced 2.5 MWs wind turbine, which is under development. It has already passed prototype test and set out to be produced and installed for a number of wind energy companies in China. With some important aspects, the company is considered as an ideal organisation for industrial investigation of this research.

The wind turbine company is a typical design and development company dealing with innovative and promising products. Dating back to early 1990s, the company was set up by a group of researchers and engineers from an academic institute in the south of Germany. Their research background differentiates them from other companies, which forms a culture of focusing on research and development rather than production, sales and marketing aspects. As a design and development company, Vensys does not have their own factory in their organisational structure (as shown in Figure 3-1). They outsource all manufacturing work to other companies worldwide and just keep an assembly plant for prototyping and testing purposes. They have nine functional departments with around fifty five engineers for design and development of the two product lines. Most of the engineers hold bachelor or master degrees in engineering. In order to allocate as much resources as possible into product design and development, they also keep reducing unnecessary overhead as well. It can also be clearly seen from the organisational structure that seven out of nine departments work directly on product design and development, except the accounting and administration. Figure 3-2 shows the distribution of staff number in different functional departments. Ninety percent of the employment falls to engineering discipline and even the CEO is the chief engineer.

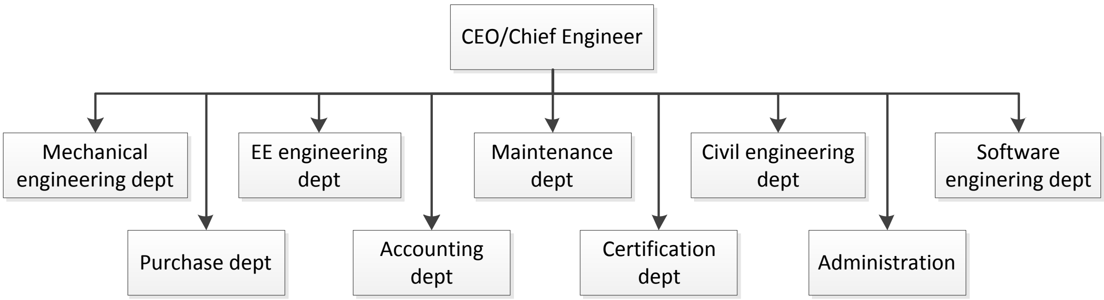


Figure 3-1 Organisational structure of the investigated company (courtesy of Vensys)

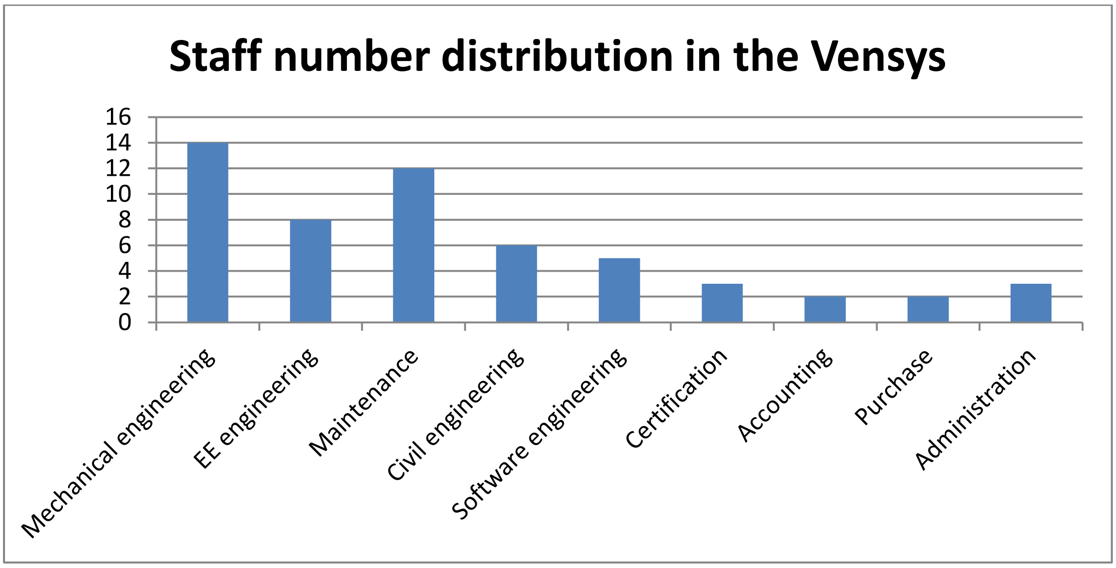


Figure 3-2 Staff number distribution in Vensys (courtesy of Vensys)

The complexity of a wind turbine makes engineering design change management very difficult in Vensys. A wind turbine is a very complex machine with more than 3000 parts, which is developed with a wide range of technologies in various disciplines (Figure 3-3 shows the inside view of the wind turbine). Most importantly it depends largely on mechanical technology to develop its giant parts with high demand of reliability. A wind turbine could reach 140 metres in height and the diameter of its blades could reach 100 metres. It could weigh 50 tons in total. For such a giant machine, its reliability is considered not just with profit and loss but also with life and death. In most common situations, a single failure of a wind turbine could cost a wind power company millions of US dollars because of the losses of electricity companies. In order to make the wind turbine work successfully, it also needs electromagnetic technology to efficiently produce electricity from wind power; electrical technology to convert current from the generator to be suitable for the power grid; electronic technology to control the blades and other parts; and software technology to monitor its working conditions all the time and control the machine remotely. For such a complex product, any engineering changes may cause significant and complex impacts on other parts of the system. However, the company does not have an efficient method and systematic process to manage engineering changes occurring in the development process. Currently, all of the engineering changes are handled conventionally by experienced engineers who have related technical skills.

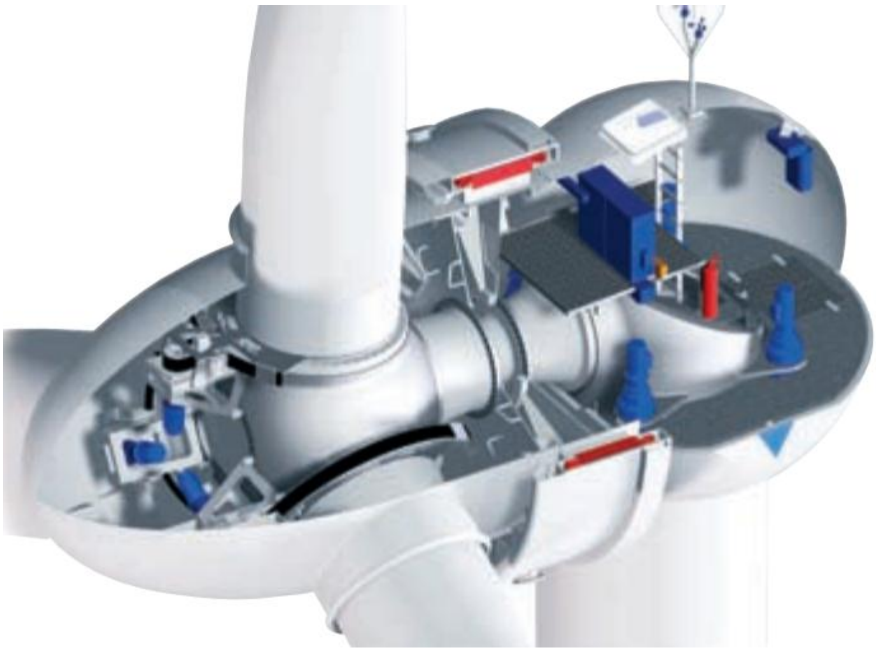


Figure 3-3 The inside view of the wind turbine (courtesy of Vensys)

3.3 Findings from the Industrial Investigation

During the industrial investigation, there are several methods used to understand the company's business and challenges they are facing. Firstly, pre-investigation is carried out by using internet technologies. The website of the company and other materials related to the company have been searched. Therefore, their business area, product lines, achievements and collaboration have been learnt. Secondly, some of the company's documents have been reviewed. Detailed product design as well as information of development processes and activities can be obtained by reviewing design documents, working regulations and communication history. Thirdly, focus group interview has been carried out. Face-to-face interviews with key members of the company are necessary in order to get implicate information from people’s experience. A questionnaire has been developed for the interviews, which mainly includes five parts, i.e. company and its internal organisational structure, product lines and collaborations, product design and development, engineering change management and knowledge management. Fourthly, filed observation is carried out to learn their products and development processes in a real world. Information obtained from document review and interviews can be materialised by watching engineers working in the field.

According to interviews with the CEO, the head of product development and the project manager of the new product, engineering design change has been considered as a big challenge in their project management since it causes a lot of uncertainties and makes the project difficult to manage, which has been leading to delays of development progress. As a SME, they have limited human resources in different technical backgrounds. Typically, one engineer may be responsible for one subsystem of wind turbine but across two product lines. For example, a mechanical engineer responsible for the wind turbine nacelle (cover housing) design needs to take care of the nacelle design of both the 1.5 MW wind turbine and the 2.5 MW wind turbine product lines. The CEO has estimated that about 60% of workforce of their product development team has been spent dealing with change requirements from the old product line (the 1.5MW wind turbine), which has severely delayed the progress of the development of the new product (2.5 MW wind turbine). That is because an engineer has to spend a lot of time on solving problems arising from engineering changes arising from the old product while he is trying to focus on the design of the new product.

Actually engineering change has been proven to be inevitable in product development. In the investigated company, the major problem, which causes delay of the new product development, is that there is a lack of effective and efficient methods to manage and solve engineering changes arising from different phases of the product life cycle, especially for changes to the product design.

By synthesising outcomes of industrial investigation from different methods, some findings have been presented in this section. Specifically for this research, the discussion of the findings has been focused on two areas: (i) engineering change management in product design and development, and (ii) knowledge management in engineering design change management.

3.3.1 Engineering Design Change Management in Product Design and Development

Figure 3-4 shows the engineering design change management process that the company adopts at the moment. The process is a kind of a stage-gate development process (Cooper and Edgett, 2006), which means that after a task is carried out, its outcome will be reviewed and decisions on the next stage of the process will be made based on the result of the review.

Engineering design changes come from different sources due to the business nature of their product. Engineering design change requests may come from customers, engineers of the company and manufacturing partners from outside. When a change request is received, the product development review board meet to decide whether the change request is worth carrying out based on the judgement from the board with their considerations of potential change impacts on the business and the product. The review board is mainly composed of management and senior engineers from related functional departments.

If the engineering design change is accepted, the task of carrying out the change will be assigned to engineers responsible for that part of the product where the engineering design change is requested or engineers in related areas and having necessary expertise. Normally one engineering change is carried out by one engineer in this company. However, depending on the importance, the complexity or the scale of an engineering change, the task may also be carried out by an engineering team. The engineers use their expertise and experience to analyse the engineering design change problem and propose candidate solutions for it.

When the solutions are proposed, the review board is called up again and review the solutions for the engineering change. In this review meeting, members of the board need to discuss the feasibilities of the solutions, their impacts on other parts of the product or impacts on customers' requirements, and then prioritise alternative solutions and choosing a most promising one. But the chosen solution is not the end. It also needs to be reviewed by their manufacturing partners to check the impacts in the manufacturing phase. If it is necessary, further work needs to be carried out to improve the solution which will be further reviewed as well. If the solution is approved by both the review board and the manufacturing partner, it will be formally documented and implemented.

However, problems are identified in the change management process and the methods that they use to solve engineering design change requests. These identified problems are discussed next.

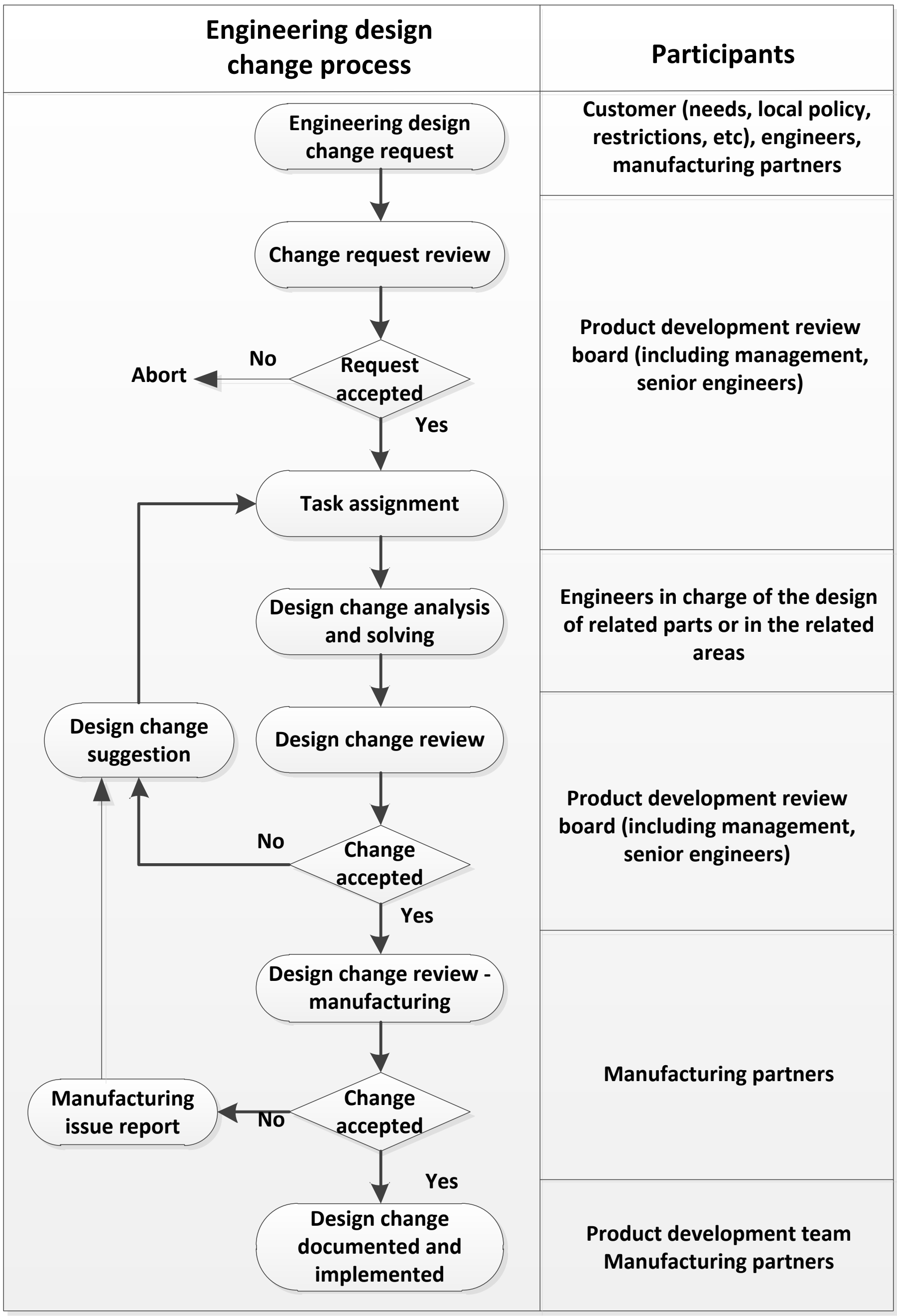


Figure 3-4 Engineering design change management process in Vensys

One of the problems found in their engineering design change management process is that there is a lack of method to analyse the impacts of changes on functional requirements and on physical components.

In terms of the products of the investigated company, changes in the functional requirement domain are normally caused by: (i) change to satisfy customer requirements; (ii) change to satisfy local government policies; (iii) change to meet implementation restrictions and; (iv) change to meet manufacturing restrictions. Any change of a functional requirement may have potential impact on other functional requirements depending on the dependency relationships among them. The dependency relationships need to be captured in the functional requirement model so that causal impacts can be analysed and controlled when engineering change resolving is being carried out. This type of impact is termed 'change impact between functional requirements (*functional-functional)'*.

It has been noted that any changes to physical components are also required to be analysed against their corresponding functional requirements. Normally, there is a solution for a functional requirement. However, in many cases, a component may get involved in more than one solution. Therefore, any change to that component may potentially have impact on the realisations of other functional requirements. This type of impact is termed 'change impact between functional and physical requirements (*functional-physical)'*.

In the current engineering design change management process of the company, impacts of functional-functional changes and the functional-physical changes are evaluated during the board review meeting. There is no systematic method for engineers to use while resolving the engineering design change. Therefore, it is difficult for engineers to consider all the potential impacts when they work on problem solving. A common situation is that an engineer assigned with a change task proposes solutions and submits to the review board for evaluation. Unsurprisingly, in most cases they are rejected, because the unexpected impacts on other functional requirements are not analysed. Therefore, the engineer needs to refine the solution according to the suggestions from the board. The refined solution may get passed or may be rejected again. This kind of iterations is very time consuming and inefficient.

The second problem identified from their design change management process is that there is a lack of method to trace change propagations. Change propagation between physical components is a recognised problem in engineering change management, which is also a main concern in the investigated company. In the current approach, an engineer does consider related components while he/she works on the solution for the engineering change. Due to the lack of a systematic method, it is difficult for the engineer to consider all possible direct change propagations, and it is even more difficult for the engineer to consider all the indirect change propagations. Without fully considering possible change propagations, the solution is very likely to be rejected during the board meeting since other engineers in the board may find other change propagations that may have serious impact on other parts of the product. Or it will be even worse if the possible change propagations are not found during the board meeting and brought into the manufacturing phase, which may cost a lot to correct. The change propagation routes within the functional domain and the physical structure domain, and the routes between them are synthesised by the author and depicted in Figure 3-5.

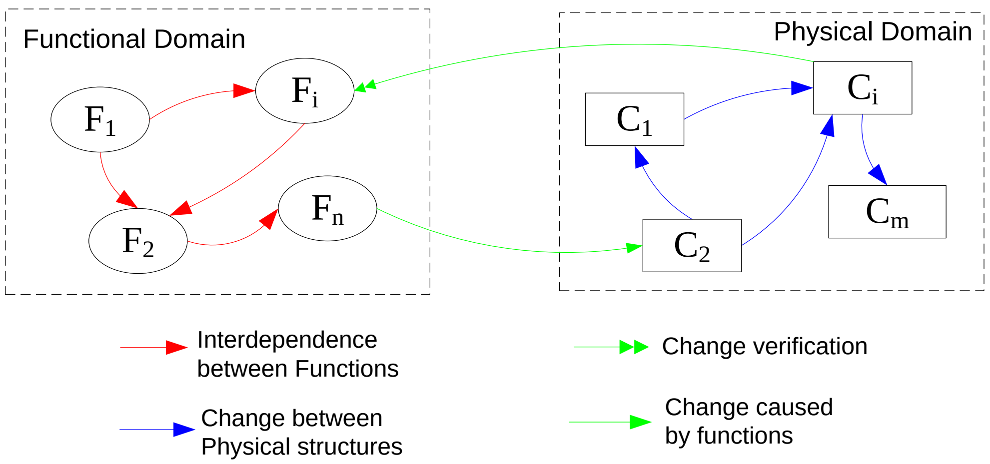


Figure 3-5 Design changes between functional and physical domains (By Author)

The third problem is that there is a lack of method for conflict resolving in design change management. As discussed above, changes of a component or a function may require other parts of the design to change. Furthermore, changes of these parts would cause changes of more parts of the design. This effect is referred to as 'change propagation or knock-on effect'. Actually, the reason why a change of a part of the design causes changes of other parts is because the initial change of the design may harm or obstruct operations of other components or satisfaction of other functional targets, which can be seen as functional or structural conflicts. In other words, change propagations are caused by design conflicts. Once there is no design conflict arising from any design changes, the change propagation stops. In the current approach, engineers have to use their experience and expertise to solve any design conflicts they find in the engineering change case. However, there are a lot of design cases that may be similar to the current problem and their solutions may be very useful. But there is no such a method in this approach that can support the reuse of previous design knowledge, which is discussed next.

3.3.2 Knowledge Reuse in Design Change Management

Knowledge reuse has been identified as an important method to facilitate product development by many researchers and practitioners. In terms of engineering design change management, knowledge reuse can help engineers find proper solutions for design change in an efficient manner and also help to make results from change analysis fact-based and consistent.

By the time this investigation was carried out, the company did not have a systematic method to manage or reuse knowledge generated from previous cases of product design and design change processing. Three problems have been identified in the company regarding knowledge management and reuse in addressing engineering design change issues.

Firstly, most knowledge generated from product design and development has not been systematically managed. Both the CEO and the head of mechanical engineering department are concerned that a method for properly managing these valuable knowledge does not exist in the company which causes massive losses. A significant part of product design and development knowledge has been stored in paper-based forms. The company has an archive room where large quantities of design documents, technical reports and sketches of conceptual ideas are stored. A lot of solutions and ideas were used to solve problems from previous design and development cases. They are proven to be effective in those design cases. However, they are not accessible by most engineers since they are not systematically managed. Also engineers do not have a way to store their solutions and ideas coming to mind while they spend a lot of time in solving current engineering problems.

Secondly, there is a lack of method to effectively and efficiently retrieve needed knowledge to solve design problems. During the company's daily operations, there are a lot of solutions that have been proposed by engineers in an attempt to tackle design conflicts arising from product development. These solutions, whether successfully implemented or just on sketches, are important assets of the company which should be properly managed and deposited in the knowledge base of the company. Once new design conflicts emerge and no similar design change mode can be referred to, engineers can follow a formalised route to try to find proper solutions for them. However, such a system does not exist yet in the company. Therefore, a common situation in the company is that engineers try to solve a problem while similar problems have been solved by other engineers in previous design cases, which is because they are not aware of those similar problems which were solved successfully. The results are that those previous solutions are wasted; delay may be caused due to time spent on the current problem; also new solutions are not proven and may cause further problems.

Thirdly, there is a lack of method to evaluate solutions for engineering change using existing knowledge in the perspective of project success. Some knowledge of physical structure development has always been learned by companies, for example knowledge regarding development time, development cost and development risks of solutions, components and parts. When a design change is initiated, the engineers not only need to find its solutions and solutions for propagating changes, they also have to estimate the overall impact caused by the initial design change by taking consideration of time, costs and risks for development of new solutions. Therefore, decisions can be made for whether it is worth going on or not, or which parts of these solutions need to be modified, in order to make sure the change impact will not be too heavy to afford. With the current approach, members of the board make a subjective judgement of the solution based on their experience.

3.4 Summary

This chapter described the investigation carried out in a typical product design and development company in the green energy industry. The company and its products were briefly introduced. It is seen that the selected company and their products are suitable for investigation for this research. Following on, the purpose and the methods employed to carry out the investigation are discussed. Finally, findings from the investigation are presented. Based on the analysis of their engineering design change management process, it is found that the company has problems in analysing and solving engineering design change while considering its impacts on realisations of other functional requirements, change propagations to other components and solving design conflict systematically. Also it is found that knowledge generated from previous design cases are not systematically managed in the company and therefore deter engineers from reusing them in solving engineering problems. It may be especially beneficial to reuse formalised design knowledge to solve design conflict occurring in engineering design change analysis.

CHAPTER 4 THE PROPOSED MODEL-DRIVEN AND KNOWLEDGE-BASED METHODOLOGY FOR ENGINEERING DESIGN CHANGE MANAGEMENT

4.1 Introduction

This chapter provides an overview of the proposed methodology and related techniques for change management in engineering product design addressing the research gaps identified in the literature review and industrial investigations. Three research gaps have been identified: (1) there is a lack of systematic method for the analysis of the impact of the changes between the functional requirements domain and the physical structure domain in the design phase; (2) there is a lack of systematic method for the analysis of the change propagation of the solutions to a design problem or conflict; and (3) there is a lack of systematic tools to help engineering designers reuse knowledge from previous cases regarding design change management in industry.

Therefore, the methodology proposed in this research needs to meet the following requirements:

- A systematic process to streamline the industrial practices of design change management.

- A systematic approach to model product design, trace design change propagations and identify design conflicts arising from them.

- A knowledge based method to semantically formalise design conflicts so that design conflicts can be reasoned and general change solutions can be found from knowledge base which semantically generalises and stores previous design cases.

- A method for evaluating change solutions in terms of their impacts on the success of the project.

In order to describe the methodology proposed in this thesis more clearly and intuitively, an example of engineering design change has been used. The industrial example used is a cooling system which is a critical part of a wind turbine described in the previous chapter. There is a real design change scenario that the wind turbine needs to be deployed in a very sandy environment so that air filtering capability of the cooling system needs to be improved to prevent more sand than in normal condition from coming into the cooling system and damaging the wind turbine. This change causes some knock-on effects that give rise to changes on other parts of the system. The proposed methodology is going to be described by using this example, which includes modelling the system, identifying design changes and related design conflicts, and resolving design conflicts using a knowledge-based system.

4.2 Overview of the Methodology

Design change management is more abstract than general engineering change management since the relationships in and between the functional domain and the physical domain should be considered. It is challenging to put these two domains together to deal with changes occurring in engineering design, especially in industry, due to a lack of support of systematic methods. Research gaps and industrial requirements have been investigated in chapter 3. In response to these gaps and requirements, a methodology is proposed to deal with change management in the engineering design phase. Figure 4-1 depicts the overview of the proposed methodology. In this diagram, the domain of the methodology is divided into five layers in order to be more understandable. The first layer reflects the structure of the proposed methodology. The proposed methodology is composed of three aspects, which includes a model-driven method for engineering change propagation analysis, a knowledge-based method for engineering design conflict solving and a method for design change evaluation. The second layer is the process of design change management. A reasonable and structural process is important to implementation of engineering design change management. In this methodology, the design change management process is composed of six phases. In each phase, there are sub-processes, methods and techniques used to solve issues of design change management. The third layer indicates the methods and techniques that are used in each phase of the process of design change management. The fourth layer shows software tools used to support the implementation of methods and techniques in the third layer. How these tools are used is critical to the implementation of this methodology.

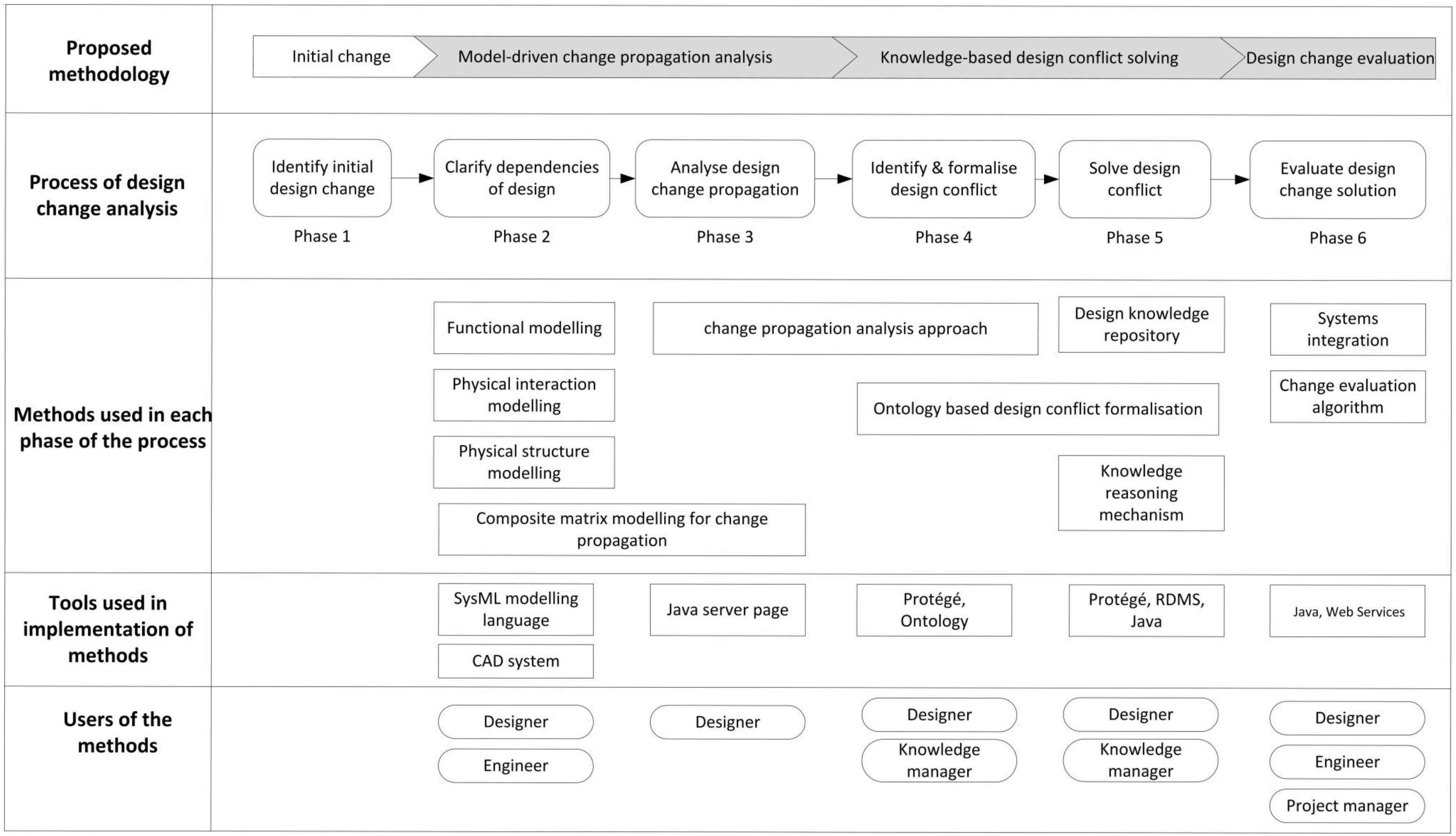


Figure 4-1 Overview of the methodology

4.3 The design change analysis process

Phase 1 of the design change analysis (see Figure 4-1) is to identify initial design changes. Initial design changes are found in this phase, which can be caused by many factors. For example, change of a functional requirement or a component can be caused by changes of customer requirements, government regulations, technical restrictions and environmental restrictions. Various reasons can trigger occurrence of a design change. The term ‘initial design change’ does not mean the first change of the design. It is used to differentiate propagated changes caused by this initial design change afterwards. Initial changes are normally identified by people involved in the project. There is no specific method proposed in this phase regarding how to identify them since they happen very randomly.

Phase 2 of the design change analysis is to clarify dependencies and relationships. It is critical to clarify dependencies and relationships existing in an engineering design, which makes change propagation analysis possible. Four types of relationships are considered, i.e., dependencies between functional requirements, involvement of physical components in realisations of functional requirements, behavioural relationships between physical components and spatial relationships between physical components. Modelling methods are used to clarify these dependencies and relationships. The SysML™ block definition diagram (Object Management Group, 2008) is used to model functional requirements and clarify the dependencies between them. The internal block diagram is used to model interactions (behavioural relationships) between physical components. CAD models are used to clarify the spatial relationships between physical components. A composite matrix model is used to summarise and simplify the dependencies and relationships clarified in those three models (i.e. block definition diagram, internal black diagram and CAD model). The composite matrix also represents mappings between functional requirements and physical components for the involvement of physical components in realisations of functional requirements. In this phase, designers and engineers are involved in the construction of these models since they know the product better than anyone else. System engineering modelling tool SysML™ and Computer Aided Design (CAD) tool are used.

Phase 3 of the design change analysis is to analyse design change propagation. Design propagation happens when a change of a design element (a functional requirement or a physical component) causes changes of other design elements (mainly physical components). It is difficult to evaluate the impact of a change if the change propagation has not been considered. During this phase, models in phase 2 are used, especially the composite matrix. When the changed components in the initial change have been identified, three types of physical components need to be analysed. The first type of components are those that work together to realise the same functional requirements. These components may need to be changed to compensate the influence caused by the changed components. The second type of components is those that have behavioural interactions with the changed components. Behavioural interactions can be represented by flows including energy flows, signal flows and material flows. The change of the component may change the states of these flows getting through it, which may further influence other components through which these flows pass. Influences on these components need to be analysed whether they can still work to meet their corresponding functional requirements. The third type of components is those that have spatial relationships with the changed component. The initial change may make the position of the component different, which may interfere with its neighbouring components that spatially connected. These neighbouring components also need to be analysed whether they can meet their corresponding functional requirements.

Phase 4 of the design change analysis is to identify and formalise design conflicts. Design conflicts happen when change of a design element harms or obstructs realisations of other functional requirements. A method for how to formalise a design conflict using domain ontology is proposed. The method for formalising design conflicts is further described in the next section.

Phase 5 of the design change analysis is to solve deign conflicts. A knowledge repository is constructed by generalising and formalising previous design cases with domain ontology. Formalised design conflicts in phase 4 are used to reason in the knowledge repository. The reasoning mechanism is developed by comparing similarities between formalised elements of design conflicts and generalised design cases. General solutions are obtained during the reasoning process. Furthermore, specific reference design cases are retrieved from database to get solutions for design conflicts.

Phase 6 of the design change analysis is to evaluate design change solutions. When solutions are obtained from the design conflicts resolving phase, their impacts in terms of factors of project success need to be evaluated. In this research, four factors are chosen to represent the extent of project success, i.e., development time, development cost, development risk and functional satisfaction. Integration with other systems such as product data management (PDM) and failure mode and effect analysis (FMEA) are developed to collect reference data for evaluation. An algorithm is also developed to calculate the final impact by considering different weights of each element of an engineering design. The detailed analytical process is depicted in Figure 4-2.

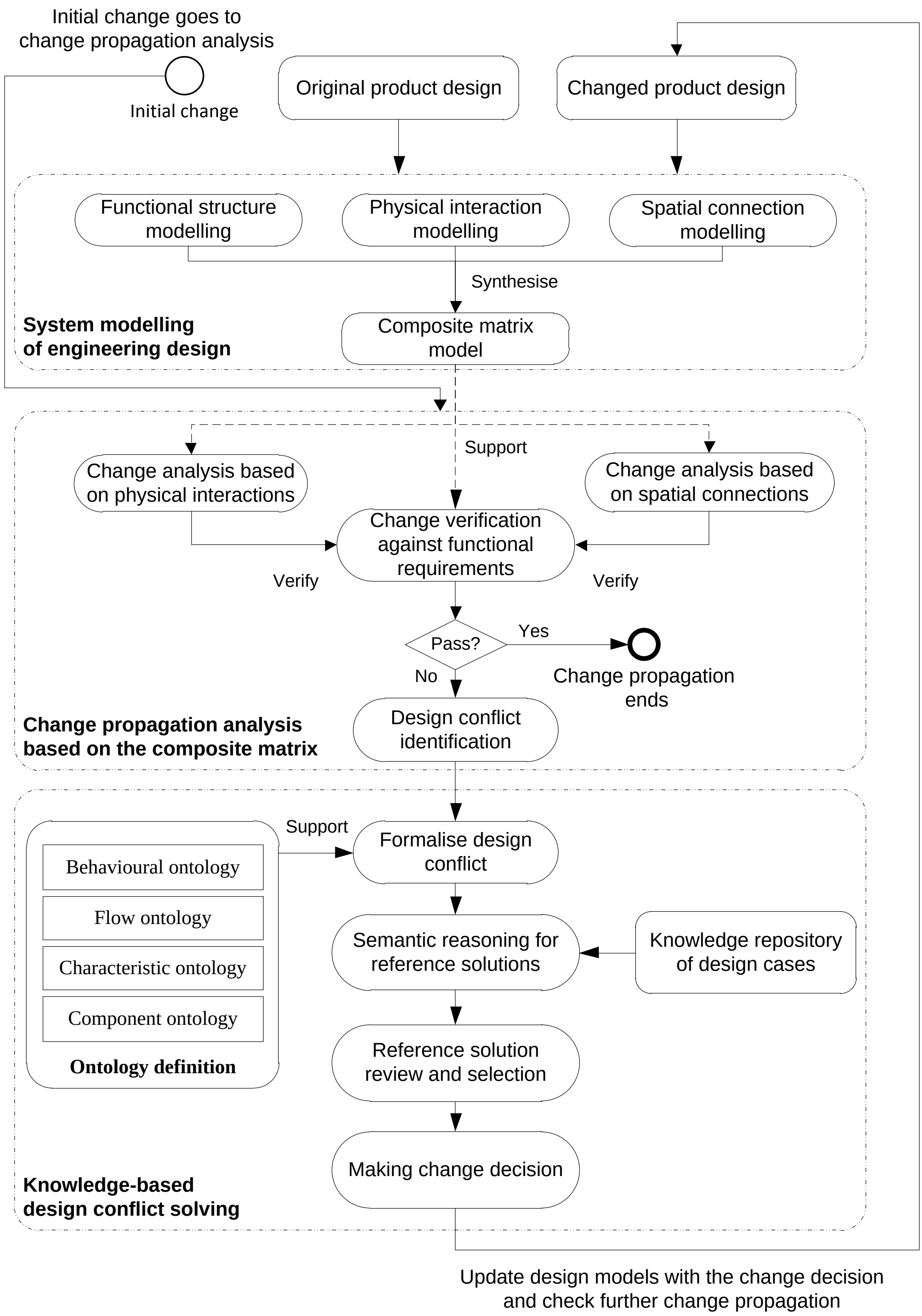


Figure 4-2 Analytical process of design change management (By Author)

4.4 The Model-driven Method for Design Change Propagation Analysis

4.4.1 Modelling methods for engineering product design

Before going into design change analysis, it is important to clarify relationships between elements of an engineering design. Elements of an engineering design that are considered in this research of engineering design change include functional requirements in the functional domain and physical components in the physical domain. Relationships of an engineering design include dependencies between functional requirements, involvement of physical components in realisations of functional requirements, behavioural relationships between physical structures, and spatial relationships between physical structures.

There are five types of models that are used in engineering design analysis. For the purpose of design change analysis, four types of models are used to clarify elements and their relationships of an engineering design which includes functional requirement model, function-components mapping model, physical interaction model and physical structure model. Correspondingly, the functional requirement model is used to clarify functional requirements and their dependencies; the function-component mapping model is for clarifying involvement of components in realisations of functional requirements; the physical interaction model is used to model behavioural relationships between components; and the physical structure model is used to model the spatial relationships between components. The other one of these five types of models is the composite matrix model. It synthesises information obtained from the other four types of models and gives engineers a composite but simplified view of the relationships between elements of an engineering design. Engineers can use the composite matrix model to analyse change propagations and design conflicts along with other four models.

Modelling methods for functional structure and physical interaction are adopted from SysML™ which is a comprehensive system engineering modelling language (Object Management Group, 2008). The reason for using SysML™ is that it is a standard modelling method having intuitively visual presentations, standard descriptive language and good software tool support. It can be easily understood by both human and computer, which is important for this project since the methodology needs to be computerised to enhance its usability. For this reason, SysML™ is better than other modelling methods in this project. Modelling of physical structure can be carried out by CAD systems to intuitively clarify the spatial relationships between physical structures. When changes of a component happen, the spatial relationship helps designers find potential changes of neighbouring components based on their positions. However, modelling of spatial relationships between physical structures is not an emphasis in this research, since a lot of research has been carried out in this aspect. The composite matrix model is proposed in this research. It is useful to intuitively display relationships between elements of an engineering design. It is a very important tool in this research to trace change propagations and help engineers find design conflicts.

4.4.1.1 Functional structure modelling

Relationships between functional requirements are modelled by the block definition diagram (BBD) of SysML™ (Figure 4-3 depicts the functional model of the cooling system of a wind turbine). The BBD is used to model the hierarchically structural relationships between functional requirements. It also helps to clarify the specifications of each functional requirement. The specification attribute of a function quantitatively or qualitatively represents what the function has to do, which is analysed by engineers from initial customer requirements or other requirements from various sources (e.g., technical restriction, management and government). Specifications are represented as attribute-value (could be precise value, value range or qualitative description) pairs. All of the sub-functions need to perform to meet their corresponding specifications so that the specifications of their parent function can be met. Any changes of specifications of functional requirements will cause consequent changes in the corresponding physical structures. In turn, any changes related to components need to be verified against its corresponding functional requirements to check whether these changes affect realisations of these functions. If functional specifications cannot be satisfied due to these changes then other necessary changes of components need to be carried out until functional specifications are acceptably satisfied.

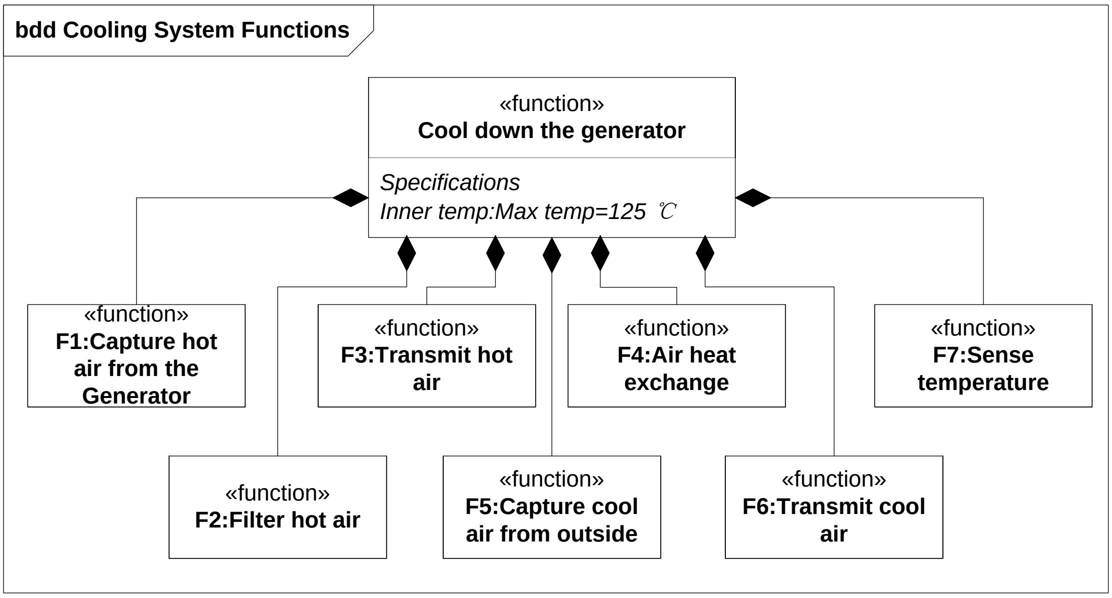


Figure 4-3 Functional analysis of cooling system (By Author, Data from Vensys)

4.4.1.2 Physical interaction modelling

Physical interaction relationship is modelled by the internal block diagram (IBD) of SysML™ to clarify the behavioural relationship between components. Figure 4-4 shows the interaction model of the cooling system. There are a variety of flows going through components, including energy flows, material flows and signal flows. A change of a component may cause changes of the flows going through it, which may also cause changes of upstream and downstream components involved in these flows. That is because the status changes of these flows may result in components not satisfying their corresponding functional requirements.

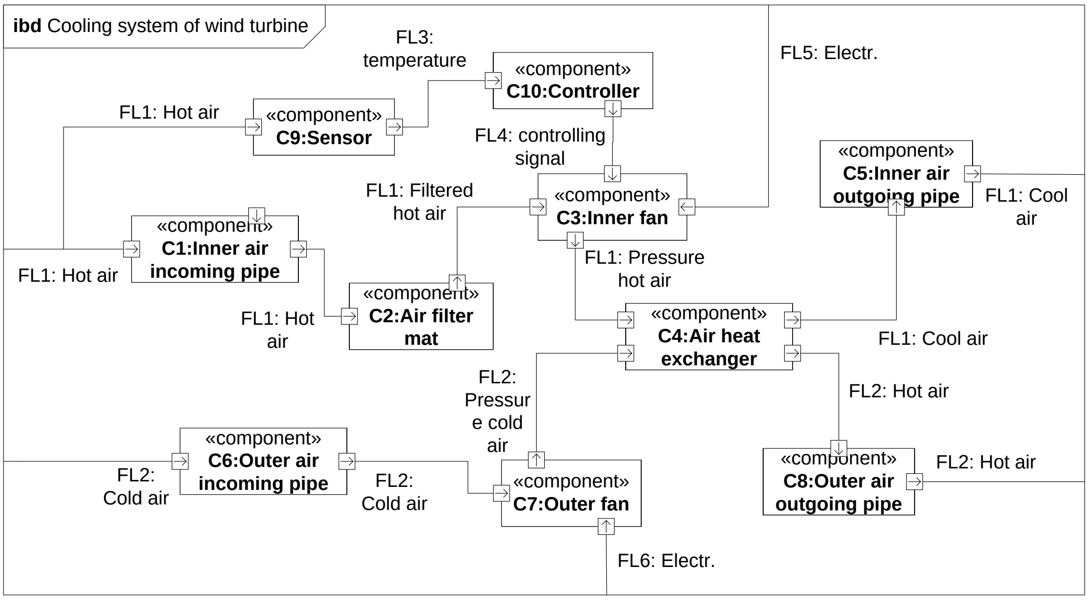


Figure 4-4 Analysis of interactions in the cooling system (By Author, Data from Vensys)

		Funtions							Flows						Components									
		F1: Capture hot air from the Generator	F2: Filter hot air	F3 :Transmit hot air	F4: Air heat exchange	F5 :Capture cool air from outside	F6: Transmit cool air	F7: Sense temperature	FL1: Air flow from the wind turbine to be	FL2: Air flow from outside for cooling	FL3: Temperature signal flow	FL4: Controlling signal flow	FL5: Electrical energy flow for the inner fan	FL6: Electrical energy flow for the outer fan	C1:Inner air incoming pipe	C2:Air filter mat	C3:Inner fan	C4:Air heat exchanger	C5:Inner air outgoing pipe	C6:Outer air incoming pipe	C7:Outer fan	C8:Outer air outgoing pipe	C9: Heat Sensor	C10:Controller
		wf(1)	*wf(2)*	*wf(3)*	*wf(4)*	*wf(5)*	*wf(6)*	*wf(7)*																
Components C1:Inner air incoming pipe	*wc(1)*	X		X					X							X	X	X					X	
C2:Air filter mat	*wc(2)*		X						X						X									
C3:Inner fan	*wc(3)*	X		X					X			X	X		X				X					
C4:Air heat exchanger	*wc(4)*				X				X	X						X				X				
C5:Inner air outgoing pipe	*wc(5)*			X					X															
C6:Outer air incoming pipe	*wc(6)*					X	X			X														
C7:Outer fan	*wc(7)*					X	X			X				X					X	X				
C8:Outer air outgoing pipe	*wc(8)*						X			X							X							
C9: Heat Sensor	*wc(9)*	X						X			X													
C10:Controller	*wc(10)*	X									X	X												

Figure 4-5 Composite matrix for design change analysis (By Author)

4.4.2 Analysis of design change propagation and design conflicts

Based on models generated as described in the above section, a method for analysing change propagation caused by an initial engineering change is described below. The scenario of an industrial application, which is described above, is used to help understand the analytical process.

An important argument brought forward in this research is that:

Change propagation is caused by design conflicts that occur when a change of a part of the system obstructs or harms realisations of functions of other parts.

Design conflicts are quite common in product development, while designers work on respective parts of a system and cannot consider dependencies between each part completely in the early phase. However, even if the system has been successfully put together, design conflicts still happen when some parts of the system change. Occurrence of design conflict has been depicted in Figure 4-6 by the author in order to help understand the idea. Given that component 2 is one of the components serving a function, when there is a change request applied to a component (component 1) which has interactional connection with component 2, it may change the input flow of component 2, which may further affect its output flow. If the affected output flow cannot satisfy the requirement of the component 2, then it is said that there is design conflict occurring at component 2 which is caused by the previous change request to component 1.

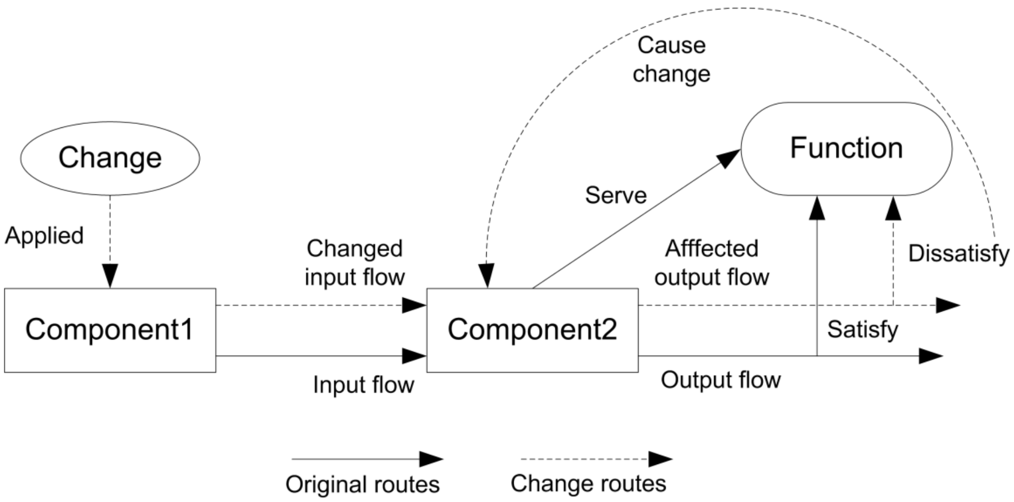


Figure 4-6 Design conflict occurring

As discussed in the literature review chapter, change propagation in engineering change management is inevitable in many cases. Although many methods have been proposed to predict change propagations in engineering change management, there is a lack of method to formalise the change propagation chain. By considering design conflicts arising during change propagations, it goes further in the analysis of change propagation based on dependencies between design elements. The change propagation process can be broken down and the impact of each phase of the propagation chain can be analysed effectively. Figure 4-7 shows the process of change propagation analysis.

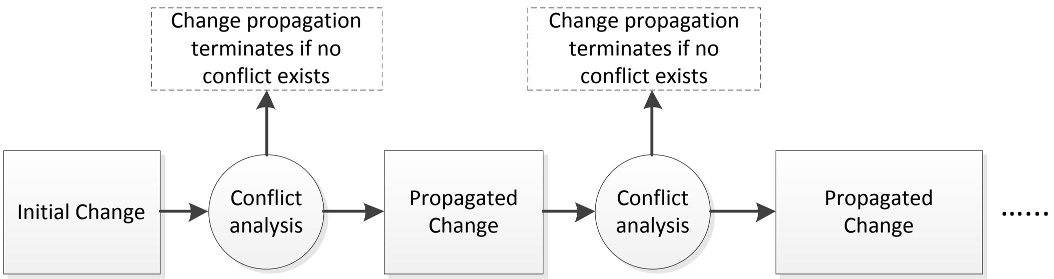


Figure 4-7 General process of change propagation analysis (By Author)

When the initial change is determined, designers need to analyse whether the result of this change may obstruct or harm realisations of functions of other parts of the system. If it does, then further changes need to be carried out. These are the so-called propagated changes. If it causes no negative influences on other parts and they can function well as designed, then change propagation ends.

Change analysis is intended to uncover changes and their propagations by following connections within functional requirements and physical components and relationships between them. The idea of analysing change propagations and identifying design conflicts arising from an initial change of a functional requirement is described below.

As discussed previously about industrial observations and analyses, there are three types of relationships existing in a product design, i.e., mapping relationship between functional requirements and physical structures, physical interaction relationship between structures, and spatial connection relationship between structures. These relationships within product design largely cause change propagations. The method of design change analysis proposed in this project is based on analyses of these three types of relationships.

The description of the method is associated with a scenario of improving air filtering as mentioned above and based on the composite matrix of change analysis (Figure 4-5). The process of identifying change propagation is described in the late part of this section. In this scenario, the change is triggered by a functional requirement called ‘F2: Filter hot air’. Therefore, the analytical process starts from the function-component part of the matrix (the green part). It is worth mentioning that the initial change can also start from the blue part or the grey part of composite matrix. If it starts from the blue part, it means the initial change is triggered by a change of a component which changes the states of flows going through it and propagates changes to its connected components and/or functions. Figure 4-8 shows the steps using the composite matrix to analyse change propagation based on this scenario.

During the proposed process of change management, knowledge of design changes is used in order to solve design conflicts arising during change analysis. Also general knowledge regarding product development is retrieved to evaluate the impact that each change may cause.

The detailed steps of change propagation analysis are described in the remaining part of this section. Descriptions of these steps are associated with the example of design change of the cooling system of wind turbine, which can help to clarify the method in a more intuitive way.

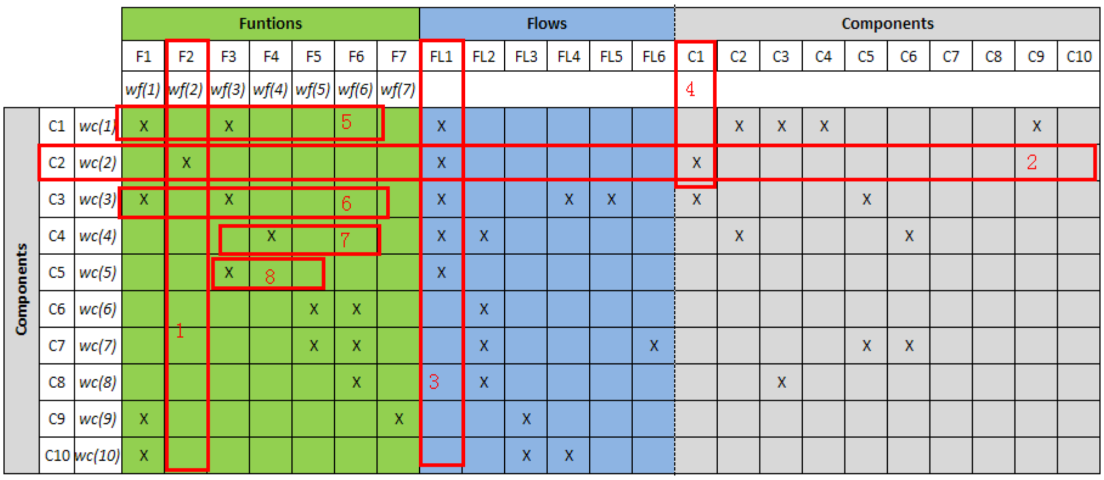


Figure 4-8 Design change propagation analysis based on simplified matrix (By Author)

Step 1: Identify component changes caused by change of corresponding functional requirement. As mentioned above, because of the sandy environment where the wind turbine will be deployed, the current air filtering measure cannot meet the new functional requirement. In Figure 4-8 Box 1 shows components involved for the realisation of the function, filter hot air (F2). In this case, there is just one component (C2, air filter mat) involved. To meet the sandy environment, the current air filter mat with a dust holding capacity of 650g/m^2 needs to be changed to a more effective one with dust holding capacity of 750g/m^2.

Step 2: Identify components that would be potentially affected by changes of components in step 1. The component changed in the above step may change the physical status of flows going through it and also it may change its neighbouring components due to changes of its spatial characteristics. Led by component C2, the row (in Box 2) shows flows and neighbouring components that are potentially affected by the change of C2. In this case, flow FL1 (air from the generator) and neighbouring component C1 (inner air incoming pipe) are related to C2. The flow FL1 also goes through C1, C3, C4, C5, so these 4 components may also be potentially affected by the change of C1. The side effects of changing the air filter mat is that the mat with higher dust holding capacity is thicker and it causes larger air pressure drop, which can significantly reduce the efficiency of heat exchange.

Step 3: Analyse change effects of each affected components and check the change effects against their related functions. Components that are affected by the flows and the spatial connections need to be checked whether the changed flows or the changed spatial connections would affect the realisations of their related functions. In this case, the air flow after the filter mat has a lower pressure which means components C3, C4 and C5 would be potentially affected since the status of air through them is changed (see the column led by FL1 in Box 3). According to the analysis by engineers, the lower air pressure through C3 (inner fan) will weaken its performance. Also the lower air pressure through C4 (air heat exchanger) will cause reduction in the efficiency of heat exchange. But it has almost no effect on C5 (the inner air outgoing pipe). The spatial change (thicker filter mat) has been considered as not notable to C1 (inner air incoming pipe) since the change can be easily accommodated by the current design. Although in this case change caused by spatial connection is negligible, in many other cases it may be significant and corresponding changes need to be made. Therefore, in this case, C3 and C4 have been identified as affected components which need to be changed to accommodate the previous change on C2.

Step 4: Identify and solve design conflicts. By analysing affected components, design conflicts can be identified. Taking C4 as an example, the changed input flow is the incoming air pressure which is lowered and the affected parameter is the heat exchange efficiency which is also lowered. This effect means the heat exchange cannot meet the functional requirement F4. Therefore this design conflict needs to be solved. In this project, a knowledge based method is developed to help designers find reference solutions from previous design cases. Detailed discussion of how to solve design conflicts using a knowledge based method is presented in section 4.5 (page 80).

Step 5: Analyse change propagations caused by component changes in step 4. When a candidate solution has been found in step 4, changes on affected components have been determined. These changes would potentially affect other components as well. In the above case, if component C4 has been changed, flows FL1, FL2 and connected components C2, C6 may also be potentially affected. Thus, a next round of change analysis also needs to be carried out until there is no further change effect being identified, which means change propagation stops and change analysis initiated by the first change is finished.

4.5 The knowledge-based method for design conflict resolving

As argued in section 4.4.2 (page 75), the reason why design changes propagate is that there are design conflicts between components when one or some of them are changed. TRIZ, originated from Russia, is a set of effective problem solving methods (Altshuller, 1996). The contradiction matrix and invention principles are useful tools of TRIZ for resolving technical conflicts. The idea of TRIZ to resolve conflicts is composed of generally four steps: (1) identify technical conflicts; (2) generalise technical conflicts by using 39 engineering parameters; (3) find invention principles via a standard contradiction matrix; (4) explore specific solutions by following the indications of invention principles (Altshuller, 1996; Fey and Rivin, 2005). Although problem solving techniques of TRIZ are innovative and inspirational to engineers, the method is difficult to master without a lot of trainings and long-time experience.

In this research, a knowledge-based method is proposed to resolve design conflicts occurring during design change propagations. It works in a similar way to problem solving with TRIZ but is more intuitive and easier by reusing previous design knowledge. When a specific design conflict is identified during the design change analysis, it will be generalised by using the functional ontology, product/component ontology and physical characteristic ontology to form a concept of design conflict. Then the generalised design conflict (the concept) will be reasoned in the knowledge base to find related knowledge which has been used or generated in previous design cases. The retrieved knowledge and its related design cases will be used as general and reference solutions to the current design conflict. The process of how design conflicts are solved is depicted in Figure **4-9**..

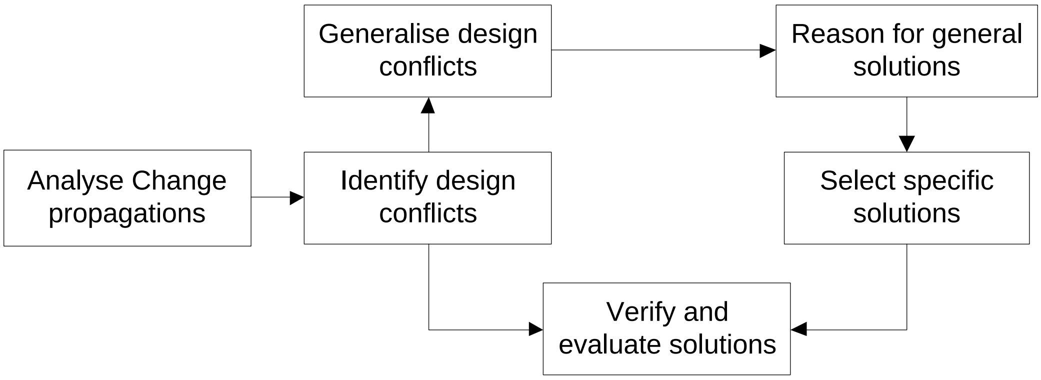


Figure 4-9 **Process of solving design conflicts (By Author)**

4.5.1 Overview of the knowledge system for conflict solving

Figure 4-10 shows the framework of the knowledge base for design conflict solving. When a design conflict arises from design change analysis, it is formalised by using the predefined to specify semantic meanings of input flows, output flows, components and their behaviours. The formalised model of the design conflict will then be input into the knowledge system. The system will reason in the knowledge repository by analysing the semantic similarities between different concepts to find most similar generalised design cases. After that, design cases associated to these generalised design cases will be retrieved as reference solutions for the current design conflicts. Designers can adjust or adopt retrieved reference solutions to solve current problems. The method of generalising design cases is as the same as the way formalising meta-interactional-model. It collects design cases and formalises their target problems or functions. The generalised design cases work as indices of those associated physical design cases.

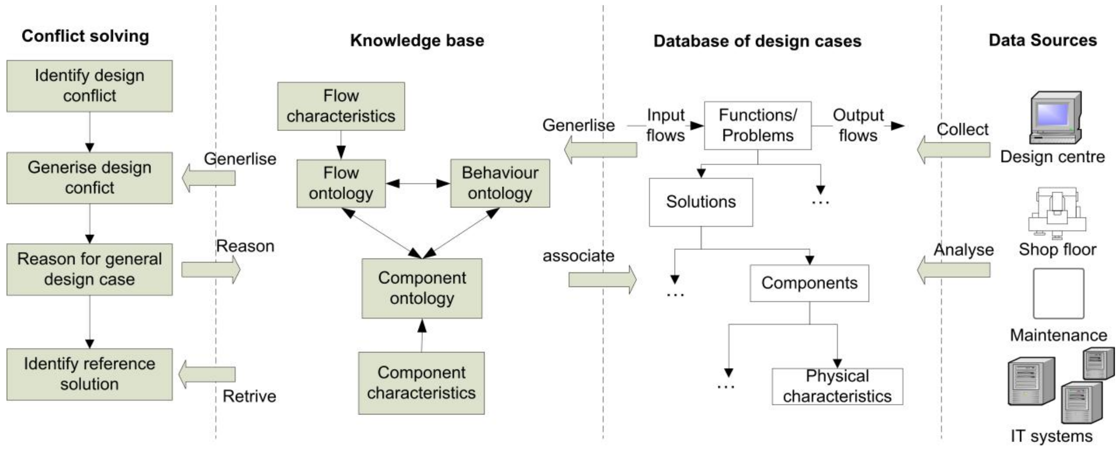


Figure 4-10 Framework of the knowledge system for conflict solving (By Author)

In this system, design cases have been collected from many sources including different functional departments and IT systems. The design cases are formalised in a hierarchical way, which clarifies the function or the problem that the design case is to address, the solutions used in this design case, the components involved in this solution and characteristics that contribute to the realisation of the function or the problem solving. The formalised structure has also been generalised by domain ontology including functional ontology, component ontology and product ontology. Therefore, the analysis of design conflict can be generalised by the same domain ontology and then general solutions for this can be found. The general solution can also be specified by following the relationship between the domain ontology and design cases.

4.5.2 Functional and component ontology for engineering products

The flow ontology and the behaviour used in this research are adopted from the work of Hirtz et al. (2002). It is called functional basis which contains generalised functions and flows which are seen as a useful and comprehensive engineering functional ontology by the author. The functional basis includes two types of ontology, namely the behaviour ontology and the flow ontology. Also the author developed domain ontology to classify components and characteristics of components and flows. Protégé is adopted as an ontology editor to develop proposed ontologies for engineering design change management. This tool which is developed by Stanford University is de facto in the academic area for ontology development. Figure 4-11 shows some parts of the ontology.

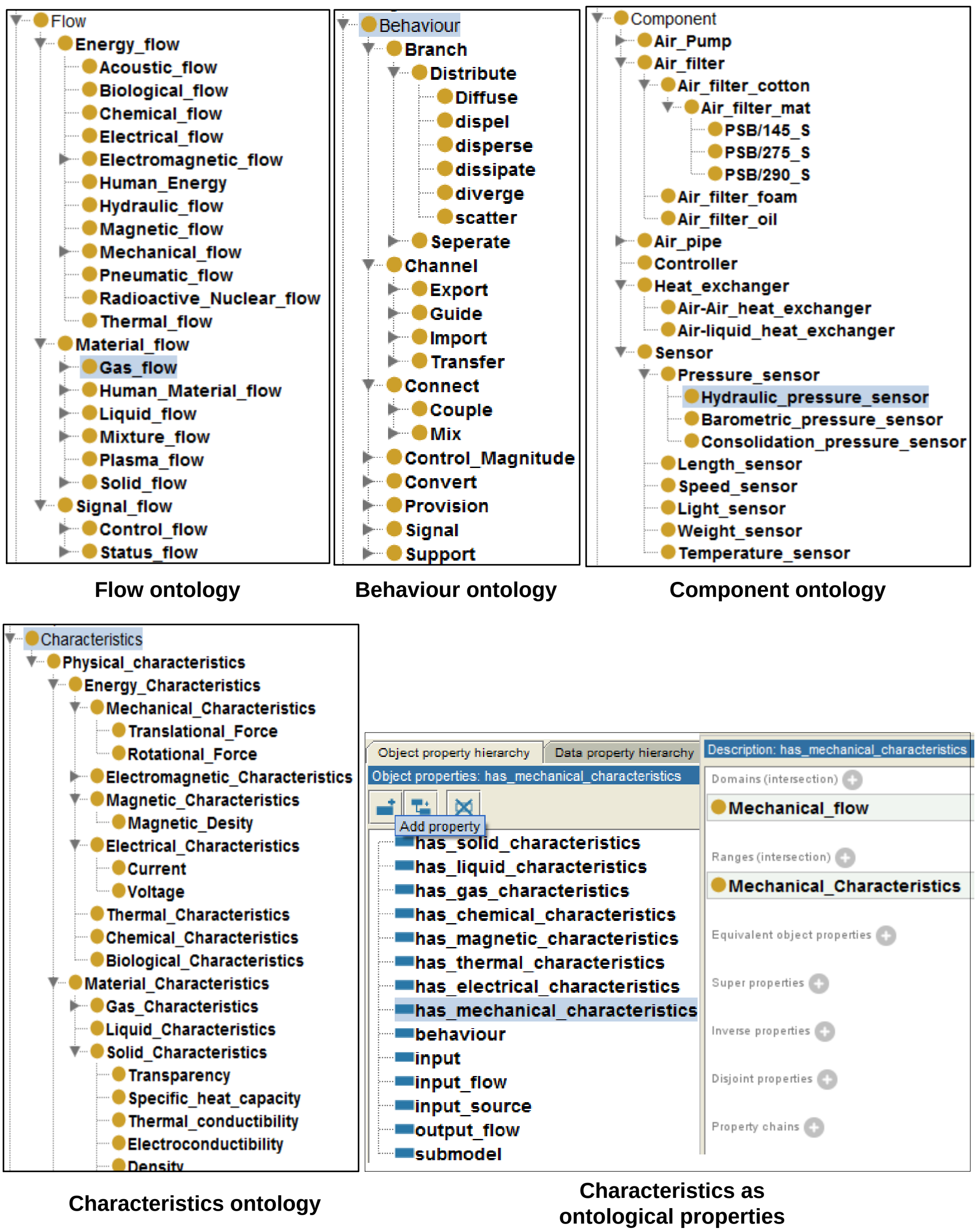


Figure 4-11 Ontology development for design change management

4.5.3 Formalisation of engineering design conflicts

In order to solve design conflicts by using a knowledge based system, design conflicts, identified through the matrix-based analysis in section 4.4 (page 70), need to be formalised with defined ontology which is the basis of the knowledge repository.

Formalising a design conflict is actually not formalising the design conflict itself. In fact, it is about formalising the interactional model (called the meta-interactional-model) of the component where the design conflict occurs. Figure 4-12 shows formalisation of a meta-interactional-model.

A meta-interactional-model includes a physical component where the conflict happens, the changed input flows and affected output flows. Both the input flows and the output flows are formalised by the flow ontology and characteristics ontology (ontology depicted in Figure 4-11). The flow ontology defines the type of flows. The characteristics ontology defines the properties of the flow. For example, the gas flow normally has properties such as pressure, temperature, moisture. Properties formalised in this part should be critical to the operation of the component. Concepts of the characteristics ontology (a node of the ontology structure) are associated with concepts of the flow ontology in the form of their properties (shown in Figure 4-11).

The component is also formalised by the behaviour ontology and the component ontology. The behaviour ontology defines what the component does with the input flows and what output flows that it generates. The component ontology defines which type of components it is. The component ontology contains related component characteristics as its properties. These component characteristics are critical for the performance of the operation of the component. For example, in the cooling system, two characteristics of the heat exchanger are important to its functionality. One is the area of the heat exchanger surface. Wider surface can have a higher heat exchanging efficiency. The other one is the material that the heat exchanger is made of. Some materials, for example bronze, have a better heat conduction performance than others such as steel.

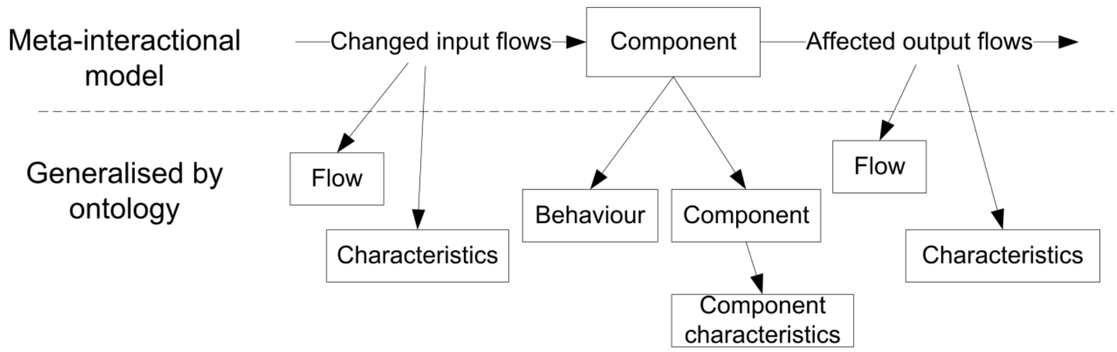


Figure 4-12 Formalisation of the meta-interaction-model (by Author)

4.5.4 Reasoning mechanism for design conflict solving

The reasoning mechanism is critical to the knowledge system for design conflict solving, since it determines the effectiveness and usability of the conflict solving method proposed in this research. The reasoning mechanism in this knowledge system involves three parts, i.e. the generalised design conflict, the knowledge repository and the semantic similarity analysis and synthesis.

The reasoning method is critical to finding candidate solutions to design conflicts arising from design change propagation analysis. It builds up the connection between generalised design conflicts and the knowledge repository to find semantically similar general solutions and then retrieve related design cases as reference solutions from the design case database. A reasoning algorithm is developed to compare semantic similarities between design conflicts and general solutions since both of them are formalised by the same set of ontology definition. The general process of design conflict solving is depicted in Figure 4-13 which is explained below in detail.

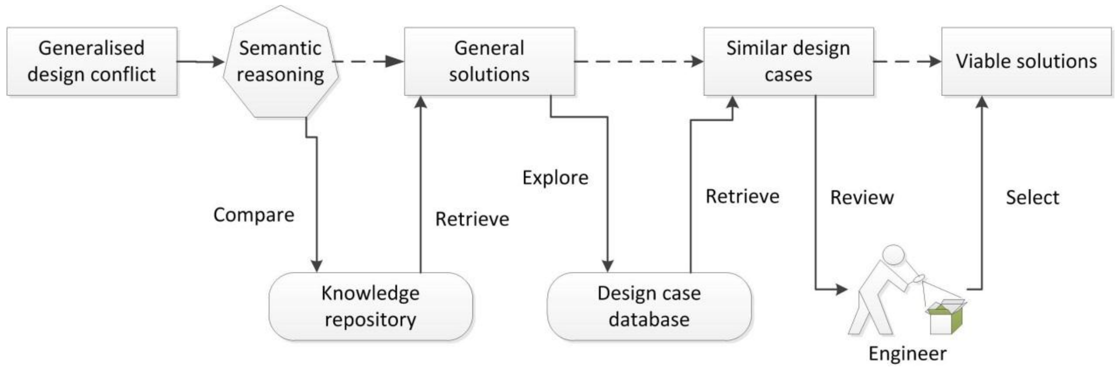


Figure 4-13 Reasoning approach to design conflict solving (by Author)

The reasoning process is composed of three steps:

(1) Finding similar general problems by using the semantic reasoning algorithm. As described above, design conflicts arising from change propagation analysis are formalised by pre-defined ontology. Problems or requirements for design cases are also generalised and formalised by the same set of ontology. Each element of the formalised design conflicts and the general problems has semantic tag attached, which makes it possible to compare the semantic similarities. Therefore, when an engineer has a design conflict to solve, he formalises it first using the proposed formalisation method and then submits the formalised design conflict to the knowledge repository. The knowledge repository employs the reasoning algorithm to calculate semantic similarities with each of the stored general problems. A prioritised list is then generated with the most semantically similar solution at the top.

(2) Selecting the most similar solutions and retrieving related design cases from the company's database. From the prioritised list generated in the last step, engineers need to select the most similar solutions and the system retrieves design cases, which are related to the selected general solutions, from the company's database. The engineer needs to find and select design cases as reference solutions which can help to solve the design conflict. Since it is not guaranteed that there would be technically suitable reference solutions, engineers may need to move further down on the list to check more general solutions with lower similarities and their related design cases, until he is satisfied with selected reference solutions. However, it is also possible that there is no suitable design case found if the design conflict is a new type which has no similar

design cases having been done before or there is no suitable design cases being formalised and stored in the database.

(3) Evaluating retrieved reference solutions and finding the most viable ones. In this knowledge-based method, reference solutions work as examples for engineers so that they can work out real solutions for the target design conflict based on information from reference solutions. However, even if real solutions are found technically viable to solve the design conflict, it does not mean they are truly viable since these solutions generated from the design change analysis need to be evaluated by some other factors which are important to the success of the product design such as development time, development cost and reliability. The evaluation method is proposed in section 4.6 (page 90).

In the proposed reasoning methodology, one of the most important steps is to analyse semantic similarities between generalised design conflicts and general solutions stored in the knowledge repository. As described above, design conflicts are generalised by predefined ontology and also previous design cases are formalised using the same set of predefined ontology. If a design conflict is defined as a concept (C) and all of the stored general solutions are defined as a set $\{C_1 \dots C_n\}$, the first step is to find the most similar solutions from the set of general solutions. Both the design conflicts and the general solutions are formalised by the same set of ontology. Each element of the design conflict and the design case is tagged by an ontological definition of the predefined set of ontology. The algorithm for calculating the semantic similarities between a generalised design conflict and general solutions is comparing the semantic similarity of each corresponding element (e.g. the flow type of the changed incoming flow in Figure 4-11) and then adding them up to get an overall semantic similarity.

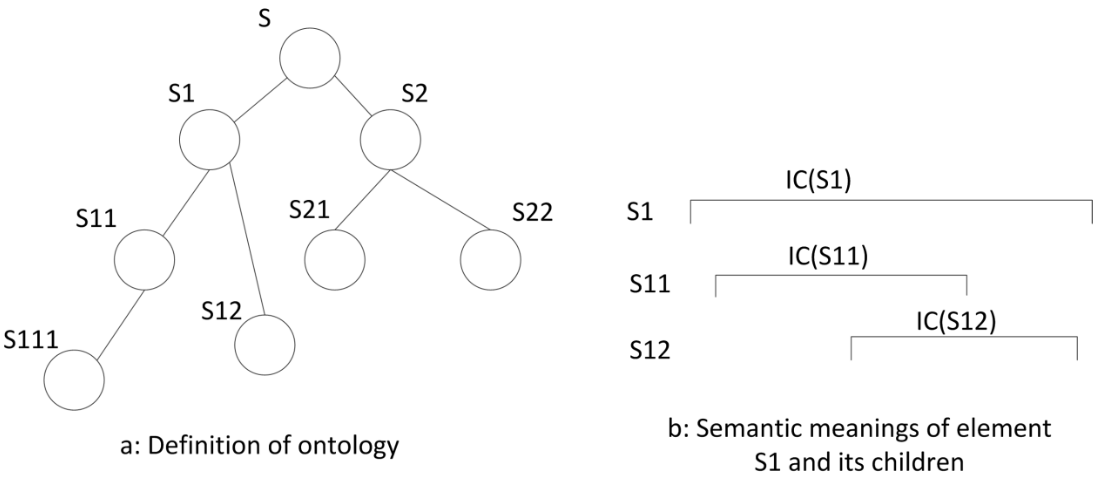


Figure 4-14 Comparison of semantic meanings between concepts (By Author)

Figure 4-14(a) represents hierarchically ontological definitions of a group of entities. The higher the level of an ontological definition, the more general semantic meaning it represents. While in lower levels, the semantic meaning of an ontological definition is more specific. In Figure 4-14(b), IC(S1) represents the semantic meaning of the ontological definition S1. Since S1 is the parent of S11 and S12, S1 has a wider semantic meaning than S11 and S12, which means:

$$IC(S11) \in IC(S1), and\ IC(S12) \in IC(S1) \tag{4.1}$$

The following equation can represent how S11 (a child) is semantically similar to S1 (a parent) by comparing scales of semantic meaning of each ontological definition:

$$Sim(S11, S1) = IC(S11)/IC(S1) \tag{4.2}$$

While the similarity of S11 (a brother) to S12 (a brother) can be represented as:

$$Sim(S11, S12) = (IC(S11) \cap IC(S12))/IC(S12) \tag{4.3}$$

Thus, the similarity of two definitions (for example, S111 and S22) can be represented as:

$$Sim(S111, S22) = \{Sim(S111, S11) \times Sim(S11, S1) \times Sim(S1, S2) \times Sim(S22, S2)\} \quad (4.4)$$

Based on the idea of calculating similarities between ontological definitions, the generalised deign conflict can be compared with general solutions in the knowledge repository, since both of them are formalised using the same set of ontology. The formalisation of general problems that general solutions are intended to solve is the same as for design conflict formalisation. So the similarity between a generalised design conflict and a general problem can be described as:

$$Sim(\mathrm{DC}, \mathrm{GD}) = Sim(CF_{DC}, CF_{GD}) \times Sim(Behaviour_{DC}, Behaviour_{GD}) \times Sim(Component_{DC}, Component_{GD}) \times Sim(AF_{DC}, AF_{GD}) \quad (4.5)$$

Where DC represents design conflict, GD represents generalised design case, CF represents general problem, CF represents changed flow, and AF represents affected flow. Similarity between changed flows can be represented as:

$$Sim(CF_{DC}, CF_{GD}) = Sim(Flow_{DC}, Flow_{GD}) \times Sim(Character_{DC}, Character_{GD}) \quad (4.6)$$

Similarity between affected flows can be represented as:

$$Sim(AF_{DC}, AF_{GD}) = Sim(Flow_{DC}, Flow_{GD}) \times Sim(Character_{DC}, Character_{GD}) \quad (4.7)$$

By comparing the overall similarities between the generalised deign conflict and general problems, a set of prioritised similarity values are generated:

$$\{Sim(DC, GD_1), Sim(DC, GD_2), \dots, Sim(DC, GD_n)\} \quad (4.8)$$

By exploring and reviewing design cases associated with general problems (corresponding to general solutions) from cases with higher priority to those with lower priority, the suitable design cases are chosen as reference solutions for the target design conflict.

4.6 The method for design change evaluation

In section 4.4 (page 70), the system-modelling-driven and matrix-based approach to analysing design change propagation and identifying design conflicts have been described. A knowledge based method has been proposed regarding how to use knowledge from previous design cases to help to find reference solutions for design conflicts and facilitate design changes to be carried out. However, there is still a lack of support for change evaluation which will be described in this section.

When the initial change and its propagating changes are determined, it is important for engineers to evaluate whether these changes and their solutions are viable in terms of project success. There are some factors to be considered when a solution has been proposed, including development time, development cost, development risk and functional satisfaction. These factors are critical to the success of a project. Without consideration of these factors, even if a solution is technically perfect it may still lead to failure of the project as a whole.

4.6.1 Evaluation criteria

Evaluation criteria considered in this research include development cost, development time, development risk and functional compatibility. These criteria are considered as critical factors for project success. For each change solution generated from methods described above, its development cost, development time and development risk are reviewed by using information acquired from related enterprise systems or technical documents. The values of these factors are compared with the original solution (before change) and also with other alternative change solutions. As another important criterion for the project success, functional compatibility means the extent to which and how much the change solution meets the original functional requirements. An algorithm has been developed and described below to calculate the overall performance of a change solution by considering those four solution evaluation criteria. The higher the score a solution gets, the more the possibility that the solution is chosen.

4.6.2 Integration with other enterprise systems

During daily production activities, manufacturing companies accumulate a lot of experience and knowledge about the development time, development cost, and development risks of parts, components and subassemblies. Mastering this type of knowledge is very important in a well-managed company. When engineers work on the product development projects they should always bear this type of knowledge in mind.

In design change management, when a solution to a design change is proven to be functionally viable, the design team still needs to examine whether it is feasible in terms of timing, cost or development risk. All these three factors are critical to the success of the project. Therefore, at this stage, the general knowledge of engineering products is retrieved.

Information used for evaluating change solutions needs to be obtained from other enterprise systems, e.g, PDM, CMS, ERP, where product information and development information are normally stored. Basic information needed for this method is as follows:

The 'part/components name' is used as an index of the data base which indicates which engineering product this entry represents.

The 'component/subassembly where used' indicates where the part or component has been used in previous products.

The 'production type' indicates how the item is produced. This field mainly has four options according to general production activities, i.e. in-house, outsourcing, supplier producing, and standard product.

The 'development time' represents the estimated development time of the part or component. The total development time can be calculated by adding up the design time, producing time, delivery time, assembly time, and so on. Different products may have different aspects the time has been spent on. For example, a totally new part may take months to produce and a standard part in stock may just take hours.

The 'development cost' indicates how much it costs to have this item available. Regularly, it includes the patent fee, the design fee, production fee, delivery fee, material cost, labour cost, and so on. The development cost differs from part to part which is similar to the development time.

The 'development risk' indicates how much the potential quality problem it will cause. In order to get the development risk of the item, the FMEA database needs to be reviewed. The designer needs to examine all the failure modes related to this item to check which one is possible to happen in the scenario of the solution. After that, the development risk can be calculated by adding up all the risk priority numbers (RPN) of the selected failure modes.

4.6.3 The Evaluation algorithm

As mentioned above, the change impact is evaluated by four criteria, namely the functionality, development time, development cost, and development risk. With help of general knowledge base for engineering products, designers can find knowledge regarding development time, development cost and development risk of pre-change components and after-change components. The formula below is used to calculate a measure for how much impact (*CI*) a design change would cause.

$$CI = \sum_{j=1}^{n} \left\{ wc_j \times \left(\alpha \left(\frac{T_j{}'}{T_j} \right) + \beta \left(\frac{CT_j{}'}{CT_j} \right) + \gamma \left(\frac{R_j{}'}{R_j} \right) \right) \right\} \times \sum_{i=1}^{m} \left\{ wf_i \times \frac{1}{FI(F_i', F_i)} \right\} \tag{4.9}$$

This formula takes all the four criteria into consideration. Parameters in this formula are explained as follows:

Integer $j \in [1, n]$ indicates all the components being changed that are caused directly or indirectly (propagated) by the initial change.

Integer $i \in [1, m]$ indicates the functions that have been directly or indirectly influenced by the initial change.

wc_j represents the relative importance of a component C_j

wf_i represents the relative importance of a function F_i

T_j represents the developing time of the pre-change component C_j

T_j' represent the developing time of the after-change component C_j'

CT_j represents the developing cost of the pre-change component C_j

CT_j' represents the developing cost of the after-change component C_j'

R_j represents the developing risk of the pre-change component C_j.

R_j' represents the developing risk of the after-change component C_j'

α, β, γ represent coefficients of developing time, developing cost and developing risk, where $\alpha + \beta + \gamma = 1$. Values of α, β and γ are determined by designers according to the project aims. For example, if the project is time sensitive then α will be assigned a relatively larger value; if the project is cost sensitive then β will be assigned a relatively large value; or if the quality of the product is critical then γ will be assigned a larger value.

Relative importance of an entity can be acquired by comparing with entities within the same domain at the same level of detail. The relative importance of a function *f* in the functional requirement domain can expressed as *wf (f)*. It is calculated by comparing other functions at the same level of detail, in terms of their contributions to the product functionality by using AHP (Analytical Hierarchy Process) method (Saaty, 1988). The relative importance of a component *c* in the physical structure domain can be expressed as *wc(c)*. Differing from functional importance, relative importance of a component is obtained by its contributions to related functions (expressed as *wcf (c, f)*) and importance of related functions. For example, C_1 in Figure 4-5 is involved in both F_2 and F_3, then

$$wc(c_i) = \sum_{i,j}\left\{wcf(c_i, f_j) \times wf(f_j)\right\} \tag{4.10}$$

wcf (c, f) is also evaluated by AHP method by comparing the contributions of different components to a function.

In equation 4.10, *F* and *F'* represent ranges of functional specifications. Each function has some outputs, and these outputs have target values or ranges which are determined during the translation from customer requirements to functional requirements by designers. When components have been changed because of the propagation of the initial change, they would possibly change the outputs of the function whose realisation involves these components. Thus, it is important to examine the actual functional outputs against the original functional specifications, which is expressed as Functional Inconsistency (FI) to indicate how different the functional outputs are from the target functional outputs. $FI(F_i', F_i)$ can be represented as the following formula:

$$FI(F_i', F_i) = \begin{cases} 0, if\ F_i' \cap F_i = \emptyset \\ \frac{F_i'}{F_i}, if\ F_i' \cap F_i = F_i \\ \frac{F_i' \cap F_i}{F_i}, if\ others \end{cases} \tag{4.11}$$

Therefore, if the actual functional outputs range is outside of the range of the functional specification, the change impact $CI = \infty$, which means that change is not successful. Designers either need to continue the physical structure change or adjust the functional requirements.

4.7 Summary

Based on the literature review and industrial investigations, it has been found that there is a lack of support for engineering change management in the design phase of product development. The methodology of design change management is intended to facilitate four business objectives of product development projects, i.e., efficiency of development, quality of product, cost of development, and knowledge reuse. The implementation of this methodology can be summarised in three stages. In the first stage, modelling methods have been used to clarify the dependency and relationships between elements of design (including functional requirements and physical components) and analyse design propagations. In the second stage, design knowledge has been used to resolve design conflicts emerging from analysis of change propagation. In the last stage, a method has been proposed to evaluate design change solutions by considering four factors of project success.

There are four types of models employed which include functional dependency model, physical interaction model, CAD model and the composite matrix model. The functional dependency model employs SysML™ block definition diagram to capture functional requirements and their relationships. The physical interaction model employs the SysML™ activity diagram to capture behavioural interactions between physical components. The CAD model is used to display the spatial relationships between physical components, which is however not an emphasis in this research. The composite matrix model synthesises information from the other three types of models. It is used to trace change propagation caused by the initial change either from the functional domain or from the physical domain of product design. The process regarding how to use the matrix to trace design propagations has been described in association with an example from industry.

It is also argued in this chapter that change propagation in design change is essentially caused by design conflicts. The reason why a change passes to another one is because the implementation of the first change causes harms or obstructs the functionality of some part of the system. That means that part of the system also needs to change in order to get it fixed. Using these generated models, change propagations arising from the initial change can be traced and design conflicts emerging from analysis of change propagations can be identified.

The idea of design change analysis and change propagation tracing remains an important unanswered question which is how to solve design conflicts arising from change propagation analysis. A TRIZ-inspired and knowledge based method has been proposed to formalise design conflicts and reusing knowledge captured from previous design cases to obtain the most semantically similar solutions as reference solutions for target design conflicts.

In order to find solutions for design conflicts in a general approach rather than a keywords-based searching approach, the TRIZ-aspired method firstly identifies key elements of a design conflict and then uses functional and component ontology to semantically formalise the design conflict. By doing this, every element of a design conflict can find a position for itself in predefined ontology.

Using the same predefined ontology, design cases from previous engineering activities can be also formalised to store in the knowledge repository which make key elements of a solution have semantic meanings and in turn these semantic meanings are associated with certain solutions. Therefore, when some elements of a design conflict find semantically similar elements in the knowledge repository, similar or potential solutions can be retrieved.

The reasoning mechanism is used to find and compare semantic similarities between elements of a design conflict and elements of solutions from design cases. By exploring the knowledge repository, the reasoning mechanism can find similar generalised design cases for target design conflict. An algorithm is used to calculate overall similarity between a design conflict and design cases.

The evaluation algorithm is proposed to find out which solution selected from the reasoning mechanism is the optimum in terms of success of a project. Technical success in many cases does not lead to success of a project, since engineers need to consider more beyond purely technical viability. They also need to consider the timeline of the project, budget for the project set by the top management and risks of solutions in the product life cycle. Thus these three aspects (development time, development cost and development risk) plus functionality consistence are used in the proposed algorithm to evaluate potential solutions. The functionality consistence represents how much the potential solution can fulfil target functions of the design. Apart from the functionality consistence, information of developing time, developing cost and developing risk need to be obtained from other enterprise systems. Therefore, system integrations are practically necessary for the evaluation method

CHAPTER 5 SYSTEM DEVELOPMENT AND IMPLEMENTATION

Design and development of a computer aided system is one of the key objectives of this research. The methodology proposed in chapter four involves complex working processes, information processing and exchange. For example, in the design change propagation analytical process, the system engineering models and the composite matrix model contain components and their relationships. If the number of components is large in some systems, the work for engineers to carry out will be very exhausting and almost impossible in some complex engineering systems. This has also been mentioned by Suh in his book regarding using axiomatic design theory to solve design coupling problems (Suh, 2001). Also in another important part of the methodology, knowledge based techniques are used to formalise design conflict problems and find semantically similar solutions from the knowledge base. Without support of information technology, it is not possible to implement this part of the methodology since the availability of knowledge based techniques largely depends on use of software tools. Therefore, computer aided system design and development is an integral part of this research in order to make the proposed methodology useful and accessible for target users, who are designers and engineers in this research project. The details of system design and development are presented and discussed in this chapter.

Firstly, the motives and the principles of developing information system for the proposed methodology are discussed. Secondly, software platform and software tools are evaluated and chosen in order to meet the requirements of the system. Thirdly, the architecture of the information system is designed and the development process is discussed. Finally, key techniques developed for the system are presented and discussed.

5.1 Issues of System Development

5.1.1 Usability of the Proposed Methodology

One of the important motives to develop a computer aided system for the proposed methodology is to increase its usability. As briefly mentioned at the start of this chapter, the proposed methodology involves some techniques and methods that are not practically useful without software tool support.

The system modelling technique, SysML™, is a modelling language which is used in this research to model the functional requirements, physical components and relationships within them. This language is derived from Unified Modelling Language (UML), which makes it easy for human reading and computer based formalisation. Although this modelling technique can be used graphically in a hand drawing fashion, it will lose its power of consistency checking, structured data organisation, ability for systematic analysis and ability to integrate with other techniques to convert its models in other forms. For example, in this methodology, one of the key steps for engineering change propagation analysis is to convert the functional requirement models and the component interaction model to a composite matrix model for propagation analysis in the next step. Without software tool support, the conversion process will be significantly time consuming and inconsistent.

For another part of the methodology, a knowledge-based method is developed to manage knowledge from daily work of designers and engineers and also to find semantically similar solutions for design conflicts identified from design change propagation analysis. The knowledge-based method needs systematic and structured management of knowledge entries and design cases which are formalised by ontology languages such as Ontology Web Language (OWL). In order to reuse the formalised knowledge, computer aided system is necessary. Also the semantic similarity comparison process involves a lot of computations. Therefore, it is not practically useful or even viable without the support of a computer aided system.

5.1.2 Automation

Automation in some parts of this methodology is also a key to its successful application in industry. One of them is the automatic transformation from the SysML™ models to the composite matrix model. The SysML™ models include the models of functional requirements and the models of interactions between components. These relationships along with the spatial relationship from CAD model need to be reflected in the composite matrix model for engineering change propagation analysis. Since changes propagate from one part of the system to another part, the SysML™ models and CAD models need to adjust correspondingly in order to maintain functional relationships, interactional relationships and spatial relationships after any changes. That means the composite matrix needs to be re-constructed after every adjustment of the SysML™ models and the CAD models. Since the iterative change propagation analysis process may cause many times of re-construction of the composite matrix, automatic transformation by using computer aided system will make a significant contribution to relieve users' efforts to maintain the relationships.

On the other hand, the computer aided system is also necessary for the method of solution exploration for design conflicts. In this method, the identified design conflicts are formalised by the pre-defined domain ontology, just as the way of formalising the design cases to be domain knowledge. In order to find the most semantically similar solutions, a huge amount of computation needs to be carried out to compare the target design conflicts with the formalised design cases in the knowledge base. The part of the method is almost not viable to carry out manually without support of computer system.

5.1.3 Integration

Another motive to develop a computer aided system is to enable integration of the proposed methodology with other enterprise application systems (EAS). There are several aspects of the methodology which need to exchange information with other systems.

One aspect of the methodology that requires integration is the construction of the knowledge repository. The knowledge repository is composed of two tiers: the abstract and formalised tier and the design case tier. Design cases are collected from daily activities of designers and engineers, which include requirements/problems and their solutions. They are then abstracted and formalised by using the pre-defined ontology so that they can be found by comparing semantic similarities with target problems. Design cases are normally managed in other systems like Content Management System (CMS), Product Data Management (PDM) system and Enterprise Resource Planning (ERP) system. The knowledge repository therefore is required to integrate with these systems to import design cases and formalise them to be knowledge entries.

The other part of the methodology that needs integration is change impact evaluation. During change propagation analysis, design conflicts are identified and solved by finding reference solutions from the knowledge repository and modifying them to meet the requirements of the problem. During this process, alternative solutions are generated. Therefore, an evaluation method is proposed to prioritise them in terms of their potential contribution to the success of the project. As described in chapter 4, there are three criteria in the evaluation method, namely developing cost, developing time and developing risk. In order to use these three criteria to evaluate alternative solutions, information from other systems is needed. For example, in terms of the developing cost, the methodology needs to integrate with the ERP system to get the information such as unit part production cost, material cost, transport cost and administration cost. Normally the information is managed in the cost management subsystem or accounting subsystem of an ERP system. Or in some companies, they use standalone systems to management manufacturing cost.

5.1.4 Collaboration

Collaboration is also one of the most important issues in engineering change management, which is also one of the most important reasons why computer aided systems are developed in industry. For example in engineering change management, change cases are requested by some designers or engineers in any stage of the product development process, while tasks of finding change solutions are assigned to other designers or engineers more often than not. Therefore, collaborations between people asking for changes and people resolving changes are necessary in order to understand the situation, find proper solutions, assess solutions and so on. Even in the same group or team, people in different roles have different expertise so that they need to collaborate with each other to find solutions for change requests. In a computer aided system, it becomes easier to define people's duties in a structured and work-flow based manner. Therefore, efficient communications and consistent information sharing can be achieved in a systematic approach.

5.2 Software Tools for System Development

System implementation of the proposed methodology involves a variety of technologies and software tools, such as system engineering modelling technology, ontology management technology and also infrastructure technologies like programming platform, frameworks and tools. In this section, related tools and technologies are introduced and reviewed.

5.2.1 System Engineering Modelling Technology

System engineering modelling is one of the key technologies used in this research to support engineering change analysis. The modelling technology needs to be able to support model functional requirements, physical components, relationships between functional requirements and physical components, and also interactions between physical components. The results of system modelling of the target engineering change case will be transformed into a composite matrix where engineering change propagation analysis is carried out and design conflicts are identified. As reviewed in the literature review presented in chapter 2, although there are several modelling approaches proposed for system modelling, SysML™ is chosen as the most suitable one for this research. It meets the major modelling requirements for the proposed methodology. In the next section, modelling tools for SysML™ modelling language are reviewed and a suitable tool is selected for this system implementation part of this research.

Visio Stencil and Template for SysML™ (Visio SysML) is a Microsoft Office Visio™ based tool, developed by Dr. Pavel Hruby, for system engineering modelling using the OMG system engineering modelling language (SysML™). Microsoft Visio™ has a long-time proven track since 1992 and is widely used both in academia and industry for diagram drawing. It provides a comprehensive range of types of drawings, including drawing for conceptual design, organisational structure, engineering design, software design, and so on. Most importantly, it provides a mechanism for customisation by designing drawing templates for specific purposes. Dr. Pavel Hruby has made use of this mechanism and developed a set of templates for SysML™. The advantages of this tool have been summarised as follows: (i) Easy to use. Due to the comprehensive graphic features in Microsoft Visio and its user-friendly interfaces, Visio SysML™ is very easy to use with drag-and-drop operations. As long as users are familiar with Microsoft Visio and the SysML™ specifications, they will be comfortable with this tool. (ii) Full support of SysML™ diagrams. The tool contains 8 templates covering the whole set of SysML™ diagrams. The block diagram covers both the block definition diagram and the internal block diagram. (iii) Free and easy to obtain. Although it has been continuously updated and maintained, this tool is free for use and easy to obtain, which may be very beneficial to researchers or SysML™ learners.

Despite of its notable advantages, there are also some major disadvantages which make it unsuitable for this research: (i) Unstructured data management. Although the Visio SysML™ provides an easy way to carry out system engineering modelling, the models are basically graphic without structured data storage. (ii) Unable to check consistency. The purely graphic models do not have connections between each other, for example functional requirements models do not have connections with the physical models which are intended to realise the functional requirements. Thus, it is not possible to check the consistency between those models. (iii) Lack of ability to integrate with other tools. The unstructured modelling data leads to lack of flexibility to integrate with other information systems. (iv) Visio dependent. Although the SysML™ template is free for use and easy to obtain, the tool purely depends on the Microsoft Visio system. So if users have no access to Microsoft Visio, it will not be possible for them to use this tool either.

Papyrus is another tool developed for SysML™ modelling. Different from the Visio SysML™ discussed above, Papyrus is developed based on Eclipse which is the open-source Java Integrated Development Environment (IDE). This tool is also open-source, which means users can obtain and use it for free. It can be installed with Eclipse in the form of a plugin. The advantages of this tool are actually derived from the features of Eclipse, which include free to use, easy to modify for specific purposes or specific requirements, easy to integrate with other software tools through the Eclipse platform, eXtensible Markup Language (XML) based structured data storage. However, this tool is still under development and does not fully support all of the diagrams according to the SysML™ specification. Also some graphic elements are missing in some diagrams. Therefore it is seen as an immature modelling tool and cannot be selected as an ideal tool for the system implementation of this methodology.

Magicdraw is seen as an advanced Unified Modelling Language (UML) tool having a lot of commercial features. It can support business process modelling, system architecture modelling, software modelling and system engineering modelling. As a commercial tool, it is well developed and implements most of the features and diagrams based on the SysML™ specification. It even has a lot of add-on features to support collaborative modelling and automation. In terms of the functionality and usability, it is a mature and fully developed tool for system engineering modelling. But there are also some aspects which are considered as not suitable for this research. Firstly, as a commercial tool, it is not open for modification or further development which is crucial for a research project since it needs to meet specific and various requirements from the research not just for now but also for its future work. Secondly, although the modelling data are stored in a structured way, the data format is proprietary and cannot be accessed and transformed by third-party tools.

Topcased is a similar tool to Papyrus for system engineering modelling, which is also open source and Eclipse-based. Topcased is developed by a community which was initially formed in France and then expanded worldwide. A lot of engineering companies have participated in the development of this tool, including some famous ones like Aurbus, Tales Group and Telecom Paris Tech. There are some advantages that can be summarised as follows: (i) Full implementation of the SysML™ diagrams.

All of the 9 diagrams defined in SysML™ have been fully implemented in Topcased. (ii) Open and structured data storage. The models are stored in purely XML-based SysML™ model files which are open for integration. It is also very important that definitions of models are simple and easy to understand, therefore easy to integrate. (iii) Consistent modelling process. Similar to Magicdraw, relationships between design elements are well maintained, for example, mapping relationships between functional requirements and physical components, flow and spatial connections between components. Therefore, the consistency between different models can be maintained and examined during the modelling process. (iv) Good community support. There is an active development team with developers and engineers who are interested in system engineering modelling and have kept updating the tool many times a year. A healthy community support is a key to the success of an open source project, which means the functionalities of the tool will be up-to-date and also users can get continuous support. (v) Free to use. It is also one of the most important considerations for choosing a tool for a research project due to budget limit.

For this research project, there are two requirements for the system engineering modelling tool which are considered as necessary. Firstly and also most importantly, it should have fully implemented the diagrams defined in the SysML™ specification. Secondly, the models can be easily integrated since the SysML™ models are required to be transformed into the composite matrix model. Besides the two necessary requirements, in terms of a software project, continuous support from the vendor will help the development of the project with lower risks and higher sustainable potentials of future work. Also the cost of the tool is also a concern of a public funded research project. With similar functionalities and performances, the tool at lower cost is better. With consideration of these factors, Topcased is selected as the best system engineering modelling tool among those reviewed in this section.

5.2.2 Selection of the Ontology Editor

An ontology editor is one of the most important tools used in implementation of the proposed methodology. The main task for an ontology editor in this system is to visually build predefined domain ontology and generates OWL-based ontology definition file which is converted and integrated with the knowledge management module of the system.

There are a lot of existing ontology editors developed by different companies or academic organisations. Actually there are so many of them that it is difficult to compare all of them one by one in this thesis. For instance, 94 ontology editors were identified by Denny (2004). Notably, some researchers have carried out surveys to compare and evaluate these ontology editors (Corcho et al., 2003; Denny, 2004; Duineveld et al., 2000; Lambrix et al., 2003). For this research, the use of an ontology editor is actually very basic: *building ontology based on OWL language specification*. This requirement can be met by many ontology editors that are currently in active development. Therefore, it is not an intention of this brief review to compare the functionalities of ontology editors. What is of most concern is the usability and ease for integration. Three ontology editors are selected for review in this section, which represent trends of ontology editor development at the time, i.e. web-based ontology editor, eclipse-plugin ontology editor and standalone ontology editor.

Knoodl is a web-based ontology editor which was developed by Revelytix, Inc but then maintained by an internet community. It supports ontology building based on RDF or OWL specifications. There are some major advantages of this ontology editor: (i) It is web-based and therefore no installation is needed. This tool is built based on the cloud computing infrastructure Amazon EC2™ and can be accessed from anywhere of the world where internet access is available. There is no complex installation to do before use as long as there is a web browser on a computer. (ii) It can support collaborative ontology building due to its web-based nature. Different users are supported to access the same ontology and collaborate to work on it, which may bring a great benefit for improving the quality of the ontology and efficiency of the ontology building. But the limitation of this tool is also very notable. Due to the restrictions of web browsers, features of ontology building, such as ability of consistency check, can be compromised. Also the dependency on internet connection and the weak stability of internet browsers sometimes may cause failures when the tool is used.

The NeOn Toolkit is an ontology building tool developed by the EC-funded NeOn project which is supported by a lot of researchers and practitioners. It is based on the Eclipse framework and supports modelling of frame-like ontologies extended by rules and modelling of DL-ontologies. Beneficial from the Eclipse platform, the tool is easy to be extended by developing plugins for special needs. But it also brings disadvantages into the tool. To use the tool, users have to build an Eclipse project first but cannot just open and edit a single file to build or change ontology. That means the migration and the integration abilities of this tool are limited. But this tool can represent the near future of ontology editor tool development due to its flexibility and extensibility.

Protégé is a well-established standalone tool for ontology editing (Lambrix et al., 2003). It is developed based on the Java platform, which means it can be run on any machine with Java running environment installed. The benefit of Protégé is not just from its well implemented features of ontology building but also from the easy-to-use user interface from its relatively long time development history. Lambrix et al. reviewed the tool and thought it could give users a good of overview and feeling of control, which is a praise of the user-friendly interface. Users can also benefit from its comprehensive document support from an active development community.

Although there may be other ontology tools that can meet the basic requirements of ontology building for this research, Protégé is still chosen to be the most suitable one by the author, because of some of its advantages such as easy-to-use interface, long time track of usability and stability, comprehensive document support and easy to integrate.

5.3 System Architecture

Figure 5-1 shows the design of the architecture of the computer aided system for the proposed methodology. The system is mainly composed of three layers in terms of the software system structure. (i) The top layer represents the system presentation including user interfaces and also system integration interfaces; (ii) The middle layer represents the system application for realisations of business logics; (iii) The bottom layer represents the enterprise data storage infrastructure.

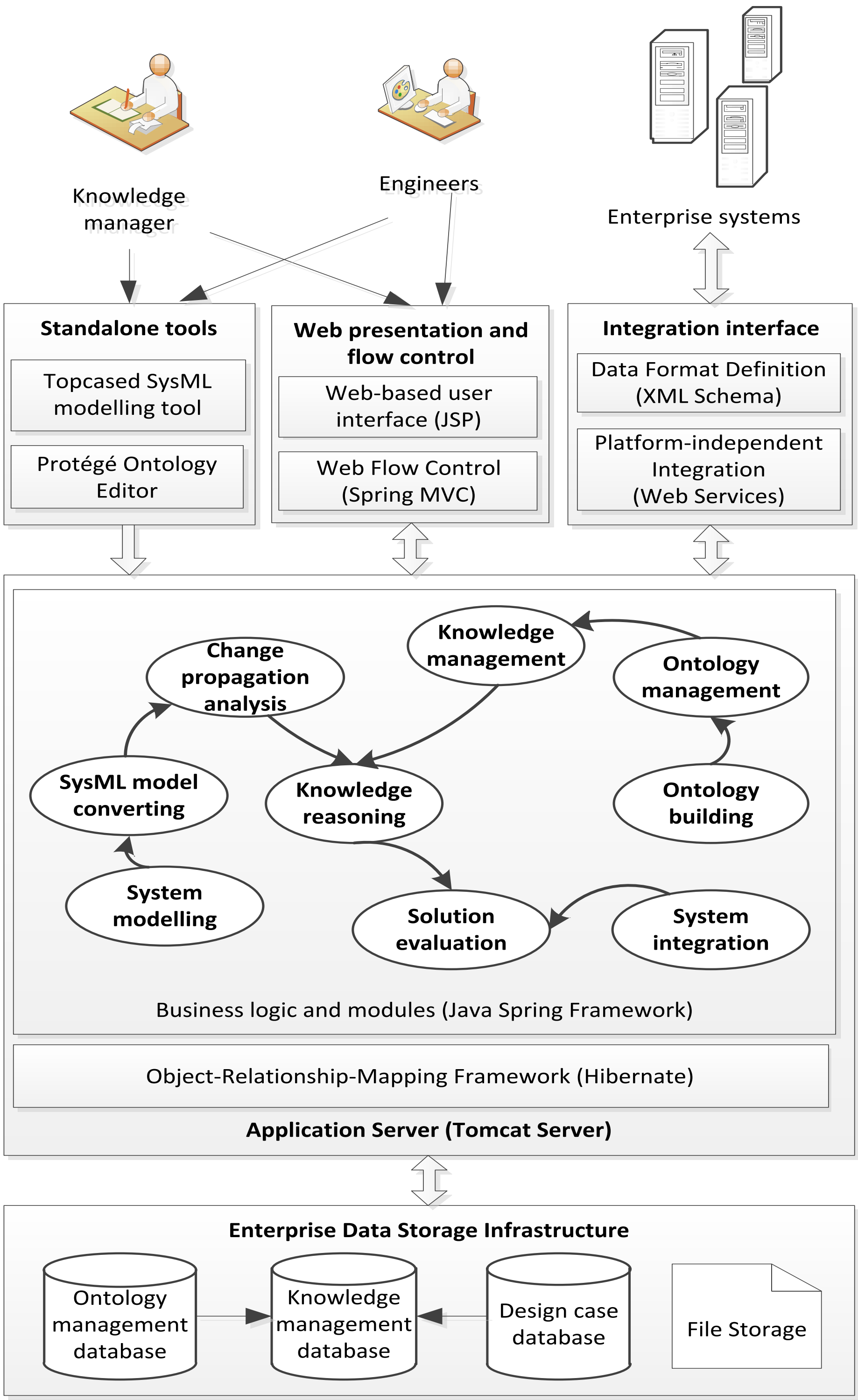


Figure 5-1 Architecture of the computer aided system (By Author)

In the top layer, there are three types of interfaces through which the computer system interacts with users and other external enterprise systems. Among them, the standalone tools include the Topcased SysML™ modelling tool and the Protégé ontology editor. The Topcased SysML™ modelling tool is used by engineers to build different models of the product design and then generate XML-based model files. These model files will be input into the system application layer and use Java XML technology to convert them into the composite matrix for change propagation analysis. The Protégé ontology editor is used by both knowledge manager and engineers to build predefined domain ontology. It generates an ontology definition file which is based on OWL2. The ontology definition file will also be converted and the concepts of the ontology will be formalised and stored in a database so that it can be systematically managed and can be used for evaluation and rating of semantic similarities by knowledge managers and engineers.

Another type of user interfaces, the web presentation and flow control module, is used by knowledge manager and engineers to interact with the system application layer of the system. This type of user interfaces use the Spring Model-View-Controller (MVC) programming model and Java server pages technology to present business data to users and also guide users to go through each step of a business operation. Most of business operations are conducted with this type of user interfaces, such as the composite matrix and change propagation analysis, semantic similarity evaluation and rating of predefined ontology, design cases and design conflicts formalisation, knowledge reasoning and solution finding, and solution evaluation.

The other type of interface is for system integration. In the change solution evaluation part of the methodology, there are different information required from external enterprise systems to evaluate the developing time, developing cost and developing risk of each proposed solution. The system integration interface is developed to communicate with those external systems and asking for relevant information. Enterprise systems like Product Data Management (PDM), Enterprise Resource Planning (ERP), Quality Management System (QMS), are developed with various technologies and based on different programming platforms, which makes it difficult for integration. To address this problem, the system integration module in this architecture uses platform-independent technologies: Web services and XML schema definition. Therefore, information stored in other system can be accessed without worries about platform restrictions. Also the format of data for exchanging can be neutrally defined by negotiations between different parties involved in the integration. Most importantly, the integration approach is loose coupled and XML-based, which makes the integration and future changes at lower costs and easier to implement, compared to hard-coded component-to-component integration.

In the middle layer, the Java Spring framework is chosen to realise core business logics. The Spring framework is a de-facto programming model in the enterprise application development area. The advantage of this programming framework includes light weight, free to use, flexibility for change, low costs for maintenance and deployment. This layer of the system implements most of important modules of the system which includes converting SysML™ models from XML-based files to composite matrix, change propagation analysis, knowledge reasoning, reference solution finding, solution evaluation, design case formalisation, knowledge management, converting ontology definition file into database, ontology management and system integration. Underneath these business logics, there is a layer for information communications between business applications and the data storage infrastructure. Hibernate, a famous Java-database mapping framework, is used in this part. With this framework, developers do not need to worry about the restrictions of particular databases while developing the system. Also developers have the freedom to switch from one database to another without troubles of modifying existing codes.

The bottom layer is the enterprise data storage infrastructure, which stores all of the data generated during the business applications described above. There are three types of databases defined in this system, namely the ontology management database, the knowledge management database and the design cases database. Also there is a file storage system designed for modelling files and ontology definition file storage.

5.4 Integration of the System Models with the Matrix Model

System models of the product design, such as the functional structure model, the physical structure model and the interactional model, are built with the SysML ™modelling tool (Topcased in this research). The SysML™ modelling tool is independent of the matrix model on which the change propagation analysis is based. As discussed in the first section of chapter 5 (page 99), automatic transformation from the system models into the matrix model is critical and manual transformation is practically not viable. Therefore, the modelling mechanism of the SysML™ modelling tool has been studied and a solution has been proposed for the automatic transformation. Figure 5-2 shows the approach which is used to automatically transform system models into a composite matrix.

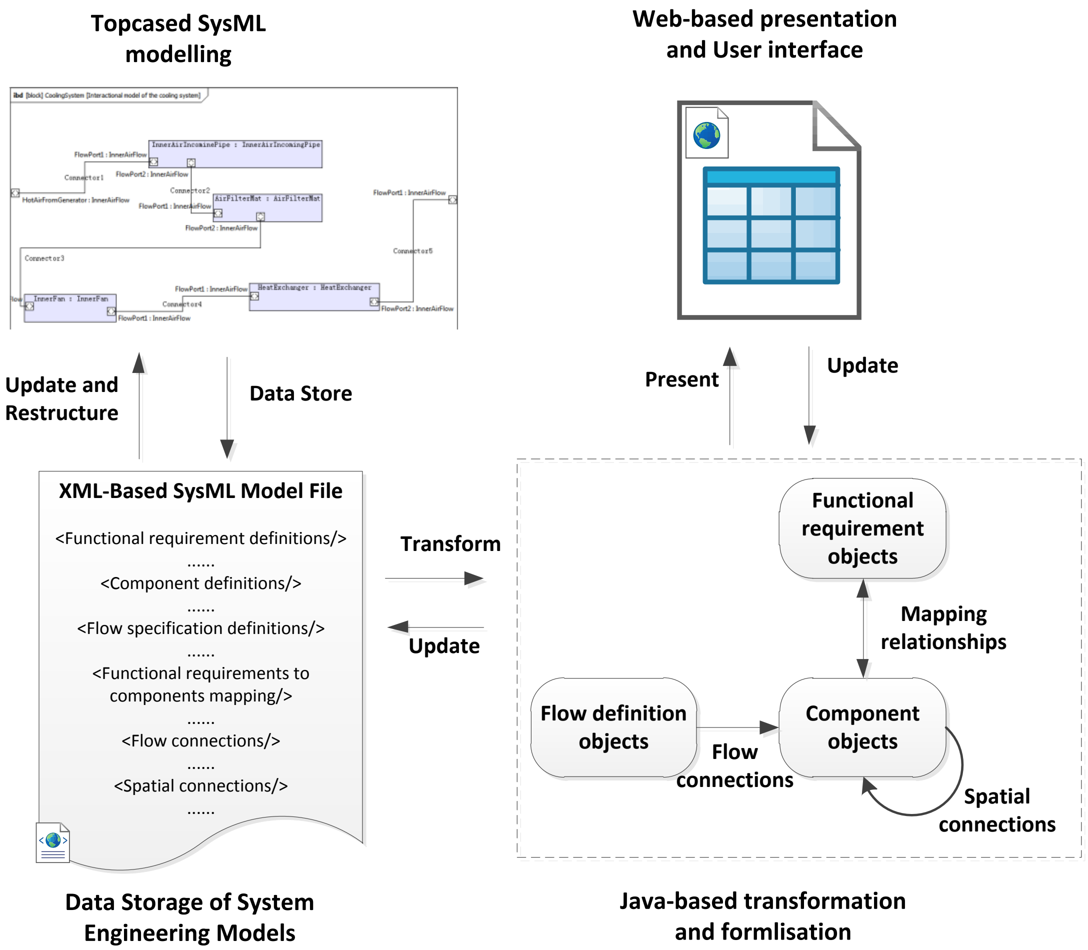


Figure 5-2 Solution for integration of the modelling tool and the composite matrix (By Author)

In this solution, the product design is modelled in Topcased with SysML™ models as described above. These models are stored in the form of a XML file. The XML file has tag definitions for each element of those models. It also uses tags to describe the flow connections and spatial connections. Then the XML-based model file is processed by a parsing program designed in the system to extract element definitions and connection definitions of the design. These elements and connections are stored in defined Java objects. The relationships between the elements and connections are maintained between these Java objects. In the end, the Java objects are populated and presented in a web page to form the desired composite matrix. The XML-based model file is shown in Figure 5-3. It is intentionally simplified for display purpose. Also a snippet of the key function "unpackModelElement" of the program which processes the model file is presented in Figure 5-4. Each part of the function has been explained with comments embedded in the codes.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<uml:Model xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore" xmlns:sysML="http://www.topcased.org/2.0/sysML" xmlns:uml="http://www.eclipse.org/uml2/3.0.0/UML" xmi:id="idModel" name="CoolingSystem">
  <!-- Indicate the model name-->
  <packagedElement xmi:type="uml:Package" xmi:id="idPackage" name="CoolingSystem">
<!--The Root element indicates the subsystem-->
    <packagedElement xmi:type="sysML:Block" xmi:id="_RK2RAFKiEeC4S6XCESBZ1A" name="CoolingSystem">
      ......
  <!-- Flow connections between components-->
      <ownedConnector xmi:type="uml:Connector" xmi:id="_JRsfQFKpEeC4S6XCESBZ1A" name="Connector2" kind="delegation">
        <end xmi:type="uml:ConnectorEnd" xmi:id="_JRsfQVKpEeC4S6XCESBZ1A" role="_9Ho9MFKoEeC4S6XCESBZ1A" partWithPort="_fn_oUFKoEeC4S6XCESBZ1A"/>
        <end xmi:type="uml:ConnectorEnd" xmi:id="_JRsfQlKpEeC4S6XCESBZ1A" role="_-44oFKoEeC4S6XCESBZ1A" partWithPort="_kZ9igFKoEeC4S6XCESBZ1A"/>
      </ownedConnector>
      ......
<!-- Components in the subsystem-->
      <nestedClassifier xmi:type="sysML:Block" xmi:id="_fr-_MFKiEeC4S6XCESBZ1A" name="Sensor"/>
<!-- Flows associated with the above component-->
        <ownedAttribute xmi:type="sysML:FlowPort" xmi:id="_B7bcUFKpEeC4S6XCESBZ1A" name="FlowPort1" type="_UYGocFKoEeC4S6XCESBZ1A" aggregation="composite" direction="out"/>
       </nestedClassifier>
        ......
  <!-- Spatial connections between components-->
      <nestedClassifier xmi:type="uml:Association" xmi:id="_G6eX8FKjEeC4S6XCESBZ1A" name="A_<innerAirIncomingPipe>_<airFilterMat>" memberEnd="_G6eX8VKjEeC4S6XCESBZ1A _G6eX9FKjEeC4S6XCESBZ1A">
        <ownedEnd xmi:type="uml:Property" xmi:id="_G6eX8VKjEeC4S6XCESBZ1A" name="innerAirIncomingPipe" type="_l4LbsFKiEeC4S6XCESBZ1A" association="_G6eX8FKjEeC4S6XCESBZ1A">
        </ownedEnd>
        <ownedEnd xmi:type="uml:Property" xmi:id="_G6eX9FKjEeC4S6XCESBZ1A" name="airFilterMat" type="_rWRcEFKiEeC4S6XCESBZ1A" association="_G6eX8FKjEeC4S6XCESBZ1A">
        </ownedEnd>
      </nestedClassifier>
      ......
        <!-- Definitions of flow types-->
      <nestedClassifier xmi:type="sysML:FlowSpecification" xmi:id="_QU0ygFKjEeC4S6XCESBZ1A" name="GasFlow"/>
      ......
      <!-- Generalisation relationships between flow types-->
      <nestedClassifier xmi:type="sysML:FlowSpecification" xmi:id="_UYGocFKoEeC4S6XCESBZ1A" name="InnerAirFlow">
        <generalization xmi:type="uml:Generalization" xmi:id="_Y2s6sFKoEeC4S6XCESBZ1A" general="_QU0ygFKjEeC4S6XCESBZ1A"/>
      </nestedClassifier>
      ......
    </packagedElement>
  </packagedElement>
</uml:Model>
```

Figure 5-3 Simplified example of the XML-based SysML model

```java
/**
 * unpack the model elements from the model file
 * store model elements in lists of components, flows,
 * spatial connections and flow connections
 */
public void unpackModelElements() {
    Element modelElement = getRoot().getChild("packagedElement")
    .getChild("packagedElement");
    List allConponentElements = modelElement.getChildren("nestedClassifier");
    List allFlowConnections = modelElement.getChildren("ownedConnector");
    for (int i = 0; i < allConponentElements.size(); i++) {
        Element eachElement = (Element) allConponentElements.get(i);
        //Find all of the components and store them in the component list
        if (eachElement.getAttributeValue("type", nsXMI)
                .equals("sysML:Block")) {
            components.add(eachElement);
        }
        //Find all of the flow definitions and store them in the flow list
        else if (eachElement.getAttributeValue("type", nsXMI)
                .equals("sysML:FlowSpecification")) {
            flows.add(eachElement);
        }
        //Find all the spatial connections and store in the spatial connection list
        else if (eachElement.getAttributeValue("type", nsXMI)
                .equals("uml:Association")) {
            spatialConnections.add(eachElement);
        }
    }
    //Find all the flow connections and store in the flow connection list
    for (int i = 0; i < allFlowConnections.size(); i++) {
        Element eachElement = (Element) allFlowConnections.get(i);
        if (eachElement.getAttributeValue("type", nsXMI)
                .equals("uml:Connector")) {
            flowConnections.add(eachElement);
        }
    }
}
```

Figure 5-4 Snippet of the model file processing program

5.5 Ontology Management

Ontology is one of the infrastructural parts of the system. It defines basic concepts used during product design development and provides a basis for knowledge management. Management of ontology mainly involves two parts, i.e. ontology defining and building, and semantic similarity evaluation and rating. Figure 5-5 shows the solution for this part of the system.

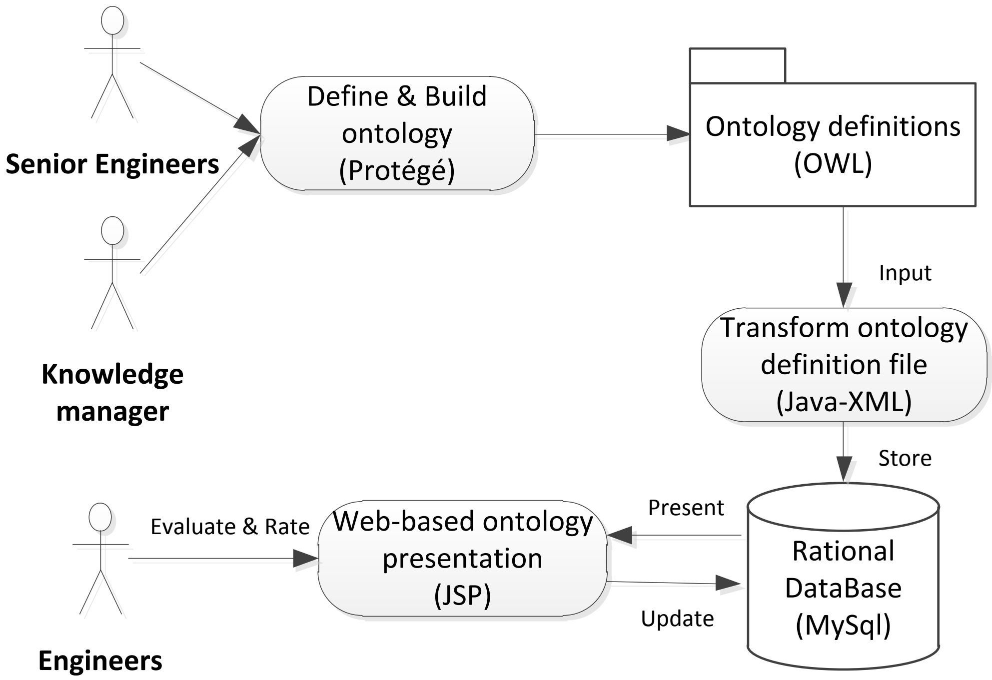


Figure 5-5 Solution for ontology management (By Author)

In the first step, the knowledge manager works with senior engineers to define concepts used in their product development process in the company. The concepts include the predefined part and the extended part. The predefined concepts are those defined in this research, which are considered as general engineering concepts (normally in higher level of abstraction). The extended concepts are those specifically used in this company, which are defined based on the predefined concepts (normally in lower level of abstraction and more concrete). They use Protégé ontology editor to build defined concepts which forms the ontology used in the system (see Figure 5-6). By using this tool, the ontology is formalised with OWL2 specifications. Physically it is stored in a XML file.

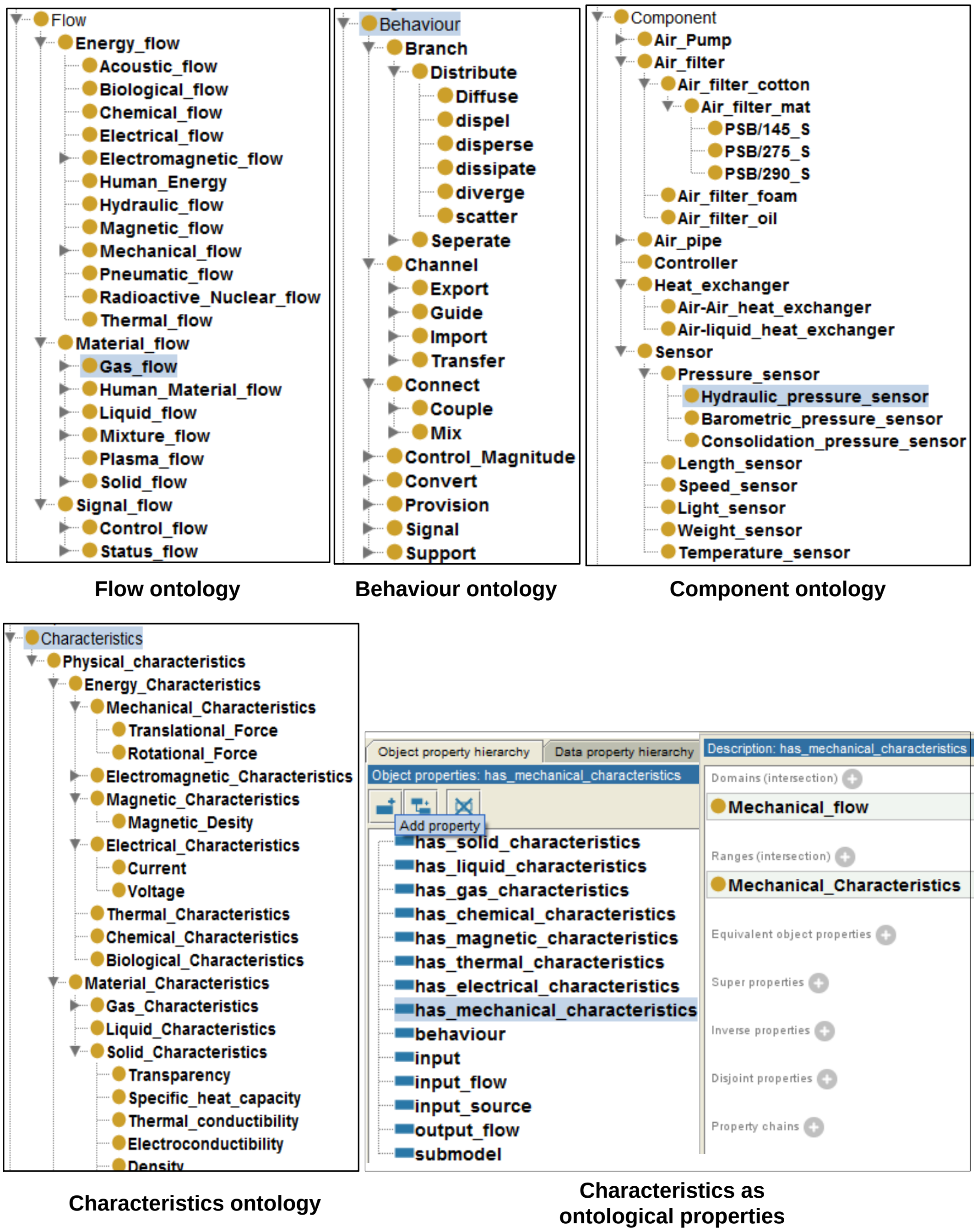


Figure 5-6 Ontology definition and building in Protégé

In the second step, the ontology definition file is processed by a program using Java technology, which is a similar way of the transformation of the modelling files. The program also stores the transformed ontology in a rational database. Therefore, it can be used with other parts of the system.

In the final step, the transformed and database stored ontology is retrieved and presented in web-based presentation technology (Java Server Page). Engineers are asked to evaluate and rate semantic similarities between concepts. The rated ontology will be used in the knowledge repository management and solution finding in the following parts of the system. Semantic similarity evaluations and rating between concepts are carried out in two steps (Figure 5-7): (i) evaluate and rate semantic similarities between the parent concept and its child concepts; (ii) evaluate and rate semantic similarities between sibling concepts.

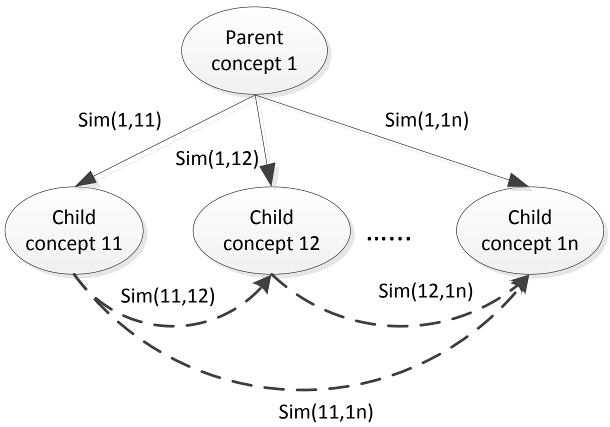


Figure 5-7 Evaluation and rating of semantic similarities of concepts (By Author)

Evaluation and rating are conducted by engineers in the company. Engineers are also rated by the knowledge manager according to their engineering experience, which means more experienced engineers have higher credibility. In other words, evaluation and rating results from experienced engineers are considered as more reliable. Taking an example of rating similarity between the parent concept 1 and one of its child concept 11 in Figure 5-7, if m engineers rate the similarity Sim(1,11) and the credibility given to each engineer is expressed as $\{C_1, C_2 \cdots, C_m\}$, then the final result of the semantic similarity Sim(1,11) will be:

$$Sim(1{,}11) = \frac{\sum_{i=1}^{m} C_i \times Sim(1{,}11)_i}{\sum_{i=1}^{m} C_i}$$

Figure 5-8 shows the implementation of concept similarity rating in the prototype system.

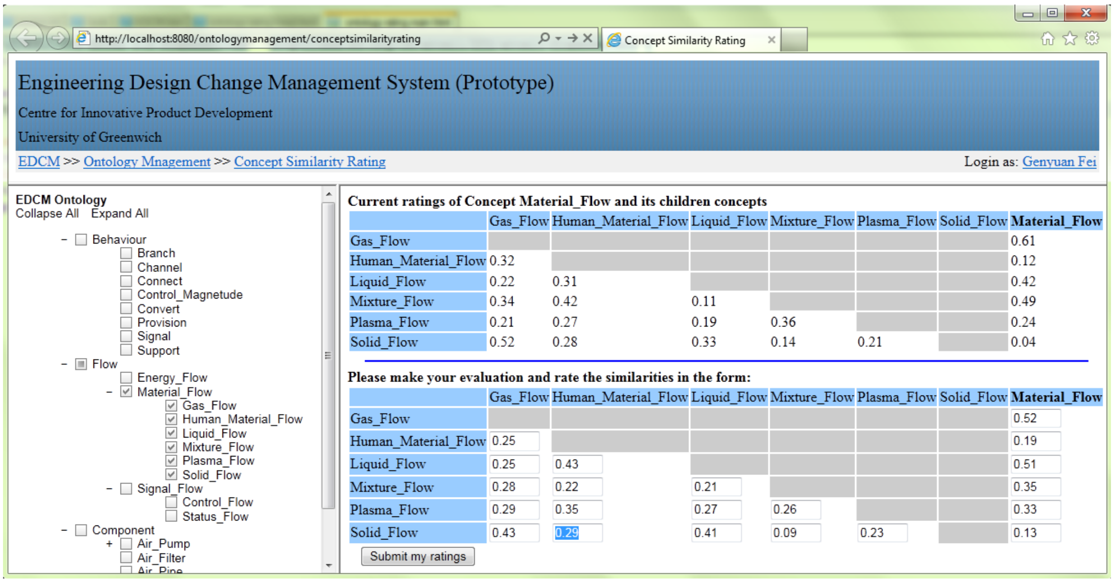


Figure 5-8 Implementation of the function of concept similarity rating

5.6 Knowledge Formalisation and Deposit

Considered as an important asset of the company, knowledge is actually accumulated from engineers' everyday work. Without an efficient and effective approach to collecting and formalising ideas and solutions from engineers' everyday work, knowledge management in the company may have no basis to stand on. An approach is developed in this research to collect and formalise design cases from engineers' work and store them as knowledge in the knowledge repository.

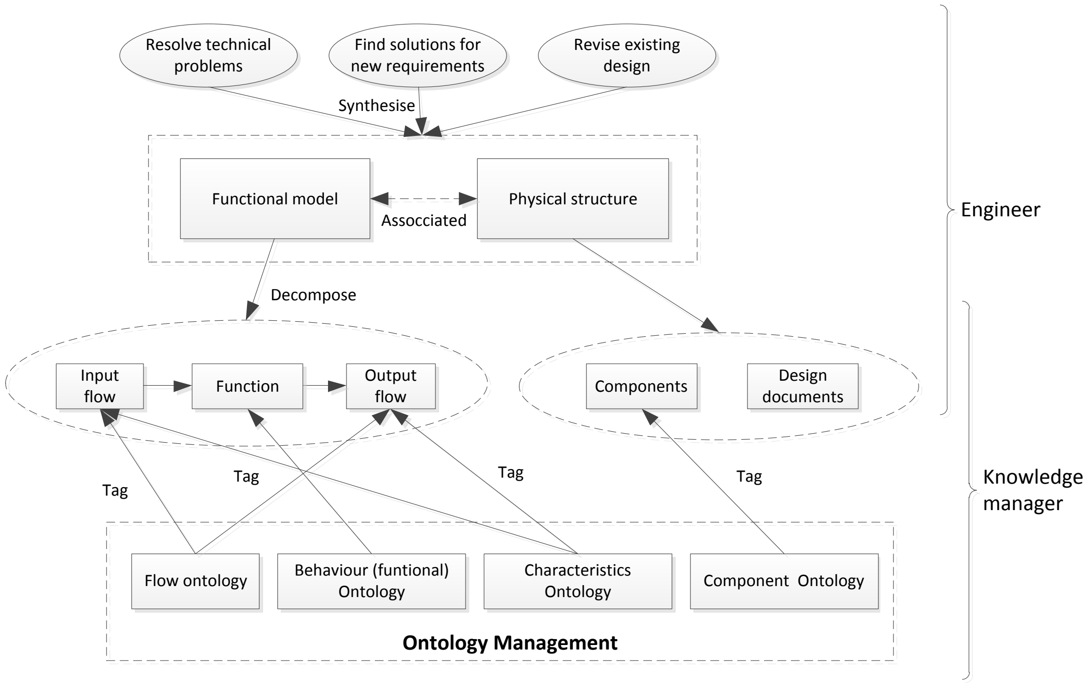


Figure 5-9 Approach to knowledge formalisation (By Author)

Figure 5-9 shows the approach to formalising design cases from engineers and store them in the knowledge repository. Design cases are collected from various activities of engineers' everyday work, but normally in three types: resolving technical problems, designing new solutions for new functional requirements, and revising existing design cases. Those design cases can be synthesised in two models: the functional model and the physical structure model. The functional model can be further decomposed into an input-function-output model. Therefore, concepts of the predefined ontology can be used to tag each element of the functional model and then the functional model with tagged semantic meaning can be stored in the knowledge repository as a knowledge entry. As depicted in Figure 4-12 in chapter 4, the input and output flows are tagged with concepts of flow ontology and concepts of the characteristics ontology to denote the types of flows and the effective characteristics that matters in the functional model. The function is tagged with concepts from the behaviour ontology to denote its semantic meaning of functionality. It may also be tagged with the object that this functional behaviour is applied to if there are more than one input flows. For example, a fan taking electricity and air as inputs is tagged with behaviour concept 'circulate' but it is only applied to the air flow. Besides the functional model, the physical structure contains structured components and unstructured design documents. The components here can also be tagged with the predefined component ontology.

In this process of transforming design cases into knowledge entries, engineers are responsible for providing the design cases and decompose them into functional model and physical structure model. Then the knowledge manager who is in charge of ontology management takes part in and works with the engineers to tag each element with predefined ontology.

5.7 Web Service Based System Integration

System integration is also an important part of this system developed for Engineering Design Change Management (EDCM). There are mainly three types of information which are obtained from other enterprise systems in order to evaluate the selected solutions for an engineering design change, i.e. the developing cost of components, developing time of components and developing risk of components. Ideally, certain information is stored in a certain enterprise system. For example, information about product should be managed in the Product Data Management system (PDM), information regarding developing cost should be managed in the Enterprise Resource Planning system (ERP). But that is not always the case since some of the functionalities of those systems are overlapping and also different enterprises may choose different information management strategy. For example, some enterprises do not implement PDM in their company therefore they manage all of the product information in an ERP system.

So it is difficult to provide integration interfaces for each information providers (enterprise application systems). In this prototype system, the technology, web services, is chosen to cope with this uncertainty of integration by providing a neutral and flexible integration mechanism as shown in Figure 5-10.

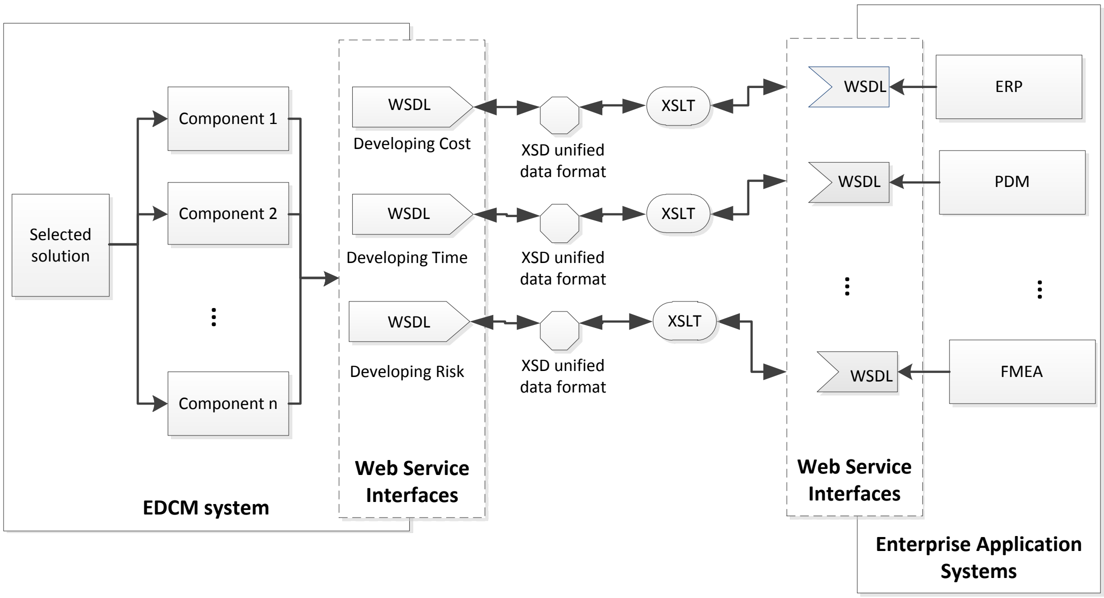


Figure 5-10 Architecture of system integration (By Author)

There are three web service interfaces developed in this system corresponding to the three types of information for integration. The backend of each web service interface is connected with the change solution evaluation module. While a component of a solution is selected for evaluation, the web services interfaces will send request to the corresponding enterprise system for certain information. The enterprise system also has its own interface to receive the request and send the information back to the EDCM system. However, in many cases, an enterprise system may need to integrate with many other systems and it does not provide specific data format for this integration operation. Traditionally, developers need to do some hard coding to implement the integration on both sides. But by using web service technology, the programs in either system do not need to change. There are two mechanisms used to ensure the flexibility of the integration. (i) XSD (XML Schema Definition) unified data definition. The web service interface is defined in form of WSDL (Web Service Description Language) in which specification of the format of information for exchange is defined in XSD. By providing unified XSD definitions for each integration interfaces of the ECDM, other enterprise systems can import the XSD definition in their web service interface. Therefore, the information format on both sides is the same. (ii) XSLT (Extensible Stylesheet Language Transformations) for data format transformation. XSLT is able to obtain XML based data from one source and transform them into another format. Therefore, if the enterprise system is unable to import the XSD unified data definition, an XSLT transformation definition file will be placed between two interfaces. Then information from the enterprise system will be automatically transformed to a format the ECDM system needs and no change needs to be made on the side of the enterprise system.

5.8 Summary

In this chapter, system implementation of the methodology, which is one of the key objectives of this research, is presented. The chapter starts with discussion of the motives to develop a computer aided system for the methodology due to its needs for improving usability, automated internal system operation, integration with internal components and external enterprise systems and also collaboration between engineers. Review and selection of system engineering tools and ontology building tools are also presented. Although there are many similar tools available in these areas, there are some special needs for the implementation of this methodology, which leads to the decision of choosing Topcased as the system engineering modelling tool and Protégé as an ontology editor.

The system architecture is also presented in this chapter. It describes the configuration of the whole system from the data storage layer to system application layer and then further to the user interface layer. Software technologies used in this development of this system are indicated in the architecture diagram. Tasks of each and functional modules layer of the system are also presented. The system architecture clearly shows the technology stack of the prototype system which well meets requirements of the computer aided system for the proposed methodology

Finally, certain key technologies which are important for the implementation of the system are described. Ideas of how the technologies work are depicted and key steps of developing those technologies are presented with diagrams, snippets or user interfaces.

CHAPTER 6 CASE STUDY AND SYSTEM VALIDATION

Based on the theoretical methodology proposed in chapter 5 and the computer aided tool developed and described in chapter 6, an industrial example is used as a case study and a walkthrough of the system with this example is presented in this chapter. From this case study, the working process with the system and its functionalities are demonstrated. This chapter also presents the validation of this system. A questionnaire was developed with consideration of criteria of system validation. Interviews with key members of the investigated company were carried out. Their comments and feedback to different aspects of the system are summarised and analysed in this chapter. Future improvements of the methodology and the system are also identified.

6.1 An Industrial Application

An industrial application is used in this project for demonstrating the developed system's working process and functionality. By walking through the system with a real industrial application, the usability of the system can be examined and drawbacks can be identified.

The industrial example is the cooling system of a wind turbine in the investigated company. A wind turbine takes energy from wind flows and converts these into mechanical forces. The mechanical forces are then further converted into electrical energy. It is a clean and sustainable energy generation technology which is seen as one of the most promising in the energy industry. Within a wind turbine, the most important subsystem is the generator which is primarily composed of a stator and a rotor. The rotor is connected with giant blades which are driven by continuous wind force. Therefore, the wind force can be converted into rotational force which is transmitted into the generator. There are coils which generate stable magnetic field around the rotor, while on the stator there are coils which are used to generate electricity. Electricity is generated by electromagnetic induction which happens while the rotor continuously spins. During this process, there is a lot of heat generated from coils from both the rotor and the stator, which is very harmful to the generator. So a cooling mechanism has to be deployed in the generator and continuously takes off the heat and reduces the temperature inside. For generators with low power rate, passive cooling mechanisms could work, for example placing radiating ribs on the surface of the stator and letting the natural wind take away the heat. But for generators with high power rate, passive cooling mechanisms are insufficient.

In the investigated company, a new active cooling system has been developed for the new product, the 2.5 megawatt wind turbine, which has a higher power rate than the old 1.5 megawatt wind turbine. The cooling system is composed of seven components, namely the air filters, the inner air pipes, the fans, the heat exchanger, the cold air inlet/outlet, the sensors, and the controller. The hot air generated from the stator-rotor area is sucked into the inner air pipe, through the air filter, by a fan which is integrated with the heat exchanger. Meanwhile cold air from outside of the wind turbine is also sucked into the heat exchanger by another fan. Within the heat exchanger, the hot air goes through a set of hot air vessels and the cold air goes through a set of cold air vessels. Each hot air vessel is placed next to a cold air vessel. Therefore, heat exchange occurs and the hot air is cooled down in this device. The cooled hot air is then pushed back into the stator-rotor area.

There is a real design change which happened while deploying the wind turbine in a wind farm. Wind turbines need to be deployed in very sandy environments where dust coming with the wind is more than normal levels. Such an environment may cause damages and then lead to unnecessary failures. In order to protect the wind turbine from such a harmful environment, the air filtering mechanism of the cooling system needs be to be improved to prevent dust from coming into the wind turbine and the cooling system. This change causes some knock-on effects that give rise to changes on other parts of the system. The methodology proposed in this thesis is going to be used to solve this problem by modelling the cooling system, identifying design changes and underlying design conflicts, and resolving design conflicts by using a knowledge-based system.

6.2 Walkthrough of the Application

Figure 6-1 shows the homepage of the engineering design change management system developed in this project. In the homepage, there are recent engineering design change cases listed. Each record of change cases shows the case number, the subsystem where the change request occurs, the component that a change is applied to, the engineer who is assigned to this change case, the current status of the change case, and a list of actions that users want to take on this change case. A user can choose to take actions such as process the change case, reassign it to someone else, or close the case. There are also hyperlinks on case numbers. By clicking them, users can review the basic information of change cases. Similarly, users can also get information regarding the engineers who are assigned with the change case and the history of the status of each change case.

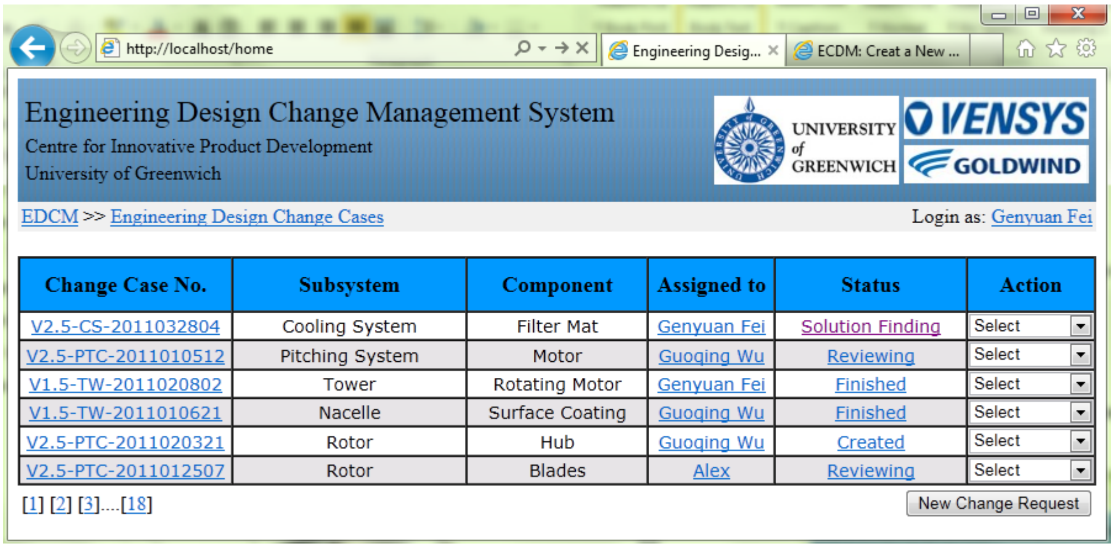


Figure 6-1 Homepage of the developed system

Figure 6-2 shows the status history of the first change case in the list. From the pop-up window which contains the status history, users can see the engineer who worked on each stage of the engineering change analysis of this case.

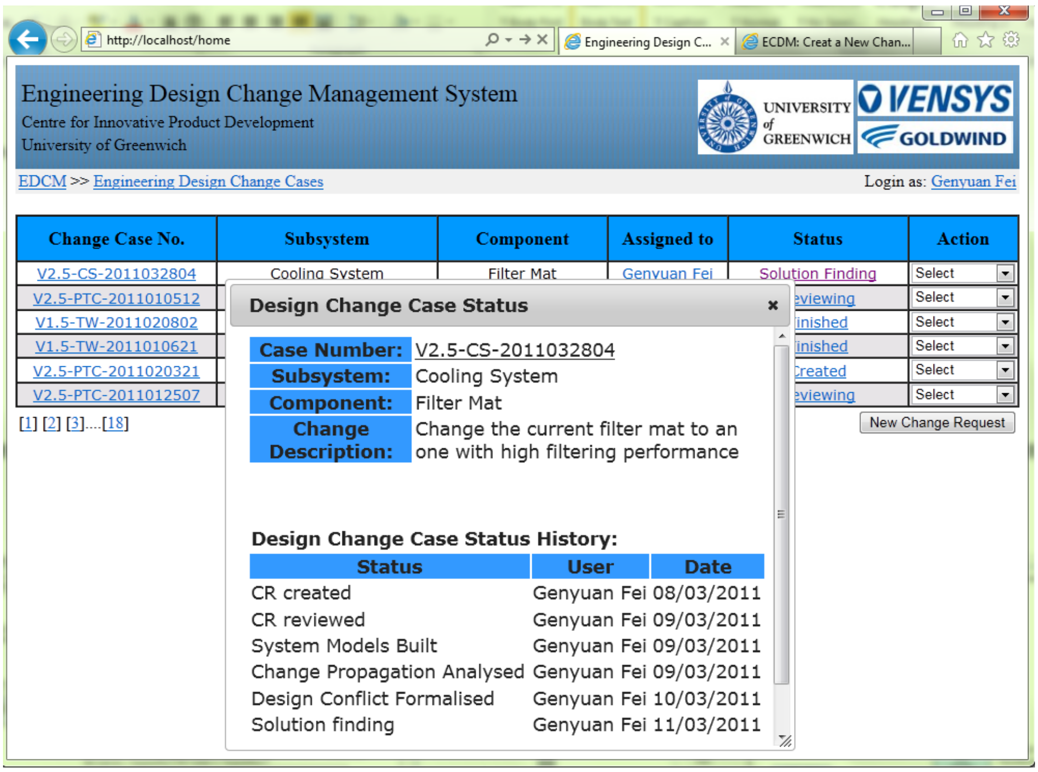


Figure 6-2 Status check of design change case

Figure 6-3 shows the interface where a new change order is created. The change order contains information such as change request number, subsystem, component, function, change request details, and change solution details. When the information is filled, the change order can be created and closed, which will be processed later. Users can choose, instead, to work straight on it and go to the next stage. In the screenshot of figure 6-3 below, a change order for changing the filtering mat of the cooling system has been created.

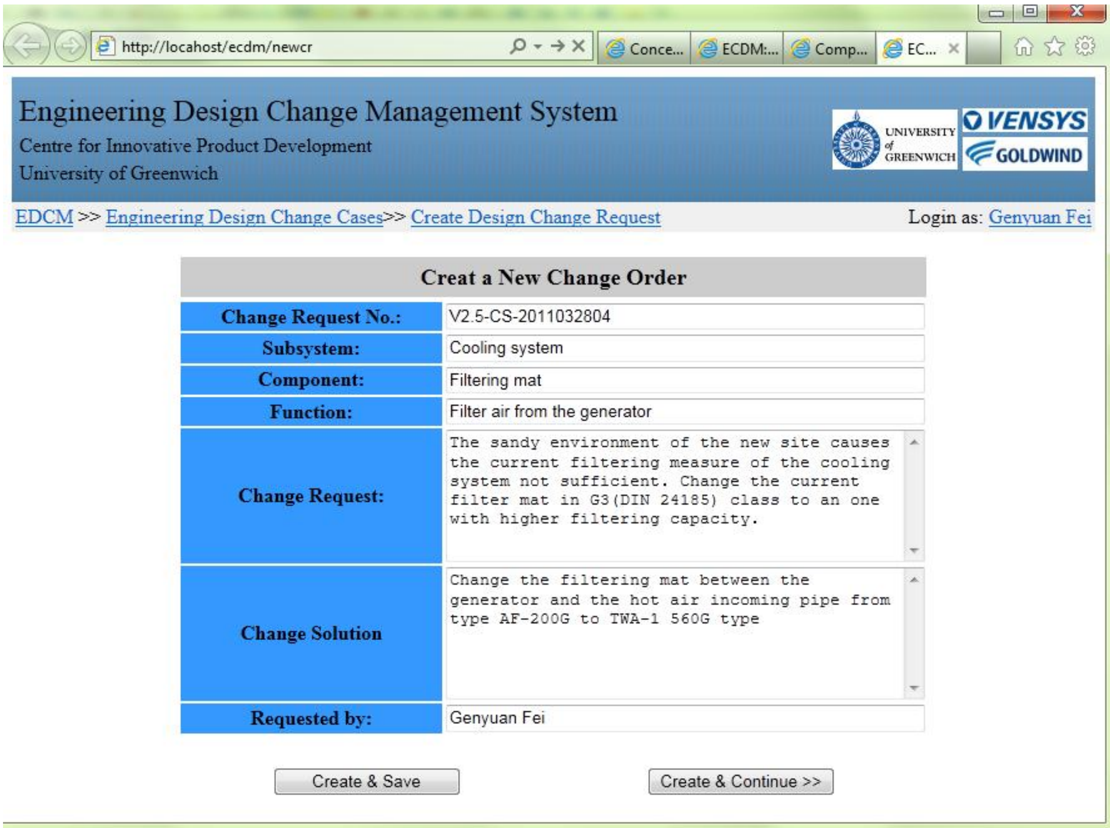


Figure 6-3 Design change case creation

Once a change order has been created, the engineer assigned with this task is to build the system models of the subsystem where the change happens. Figure 6-4 shows the interactional model of the cooling system, which is built in Topcases SysML™ modelling tool. The components and flows moving between them have been clarified.

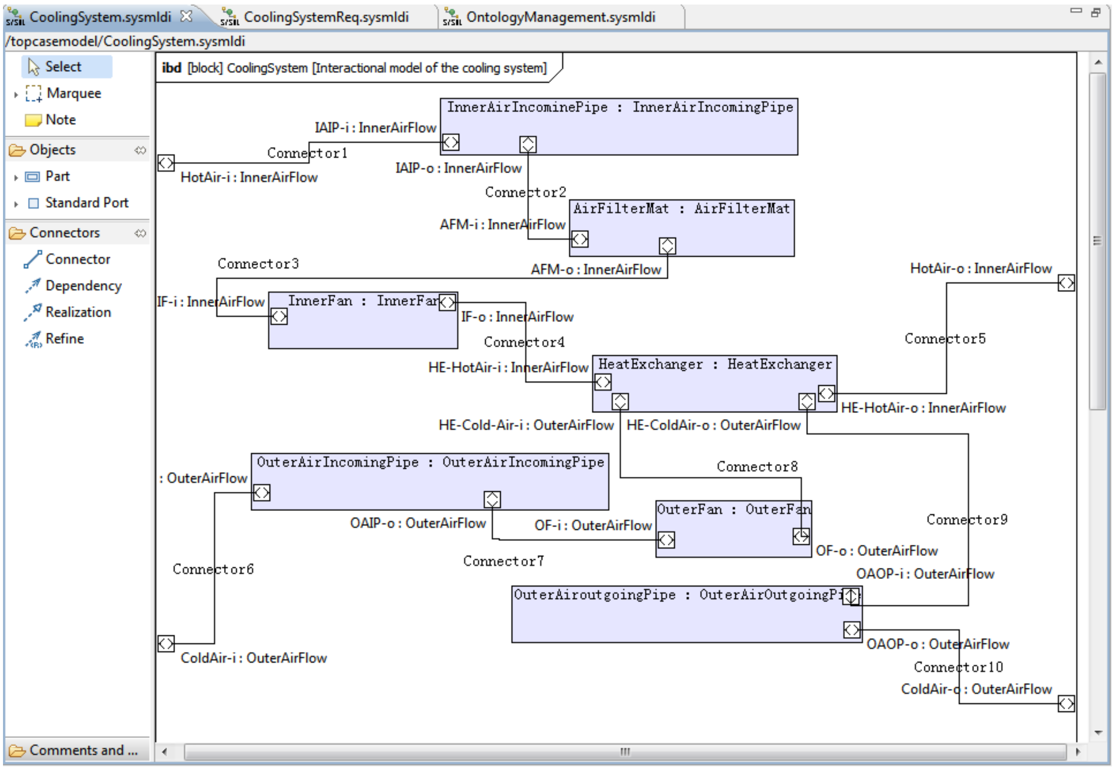


Figure 6-4 SysML interactional modelling in Topcased

Figure 6-5 shows the definitions of the flow types which are used in the interactional model.

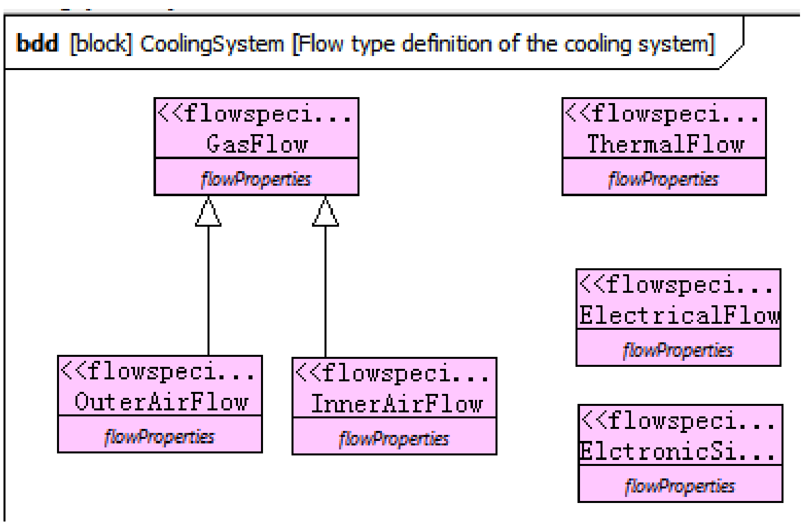


Figure 6-5 Definitions of flow types

Figure 6-6 shows the spatial connection model which presents the spatial connections between each component. The spatial connection model is manually converted from a CAD model which can clearly show the spatial relationships between components.

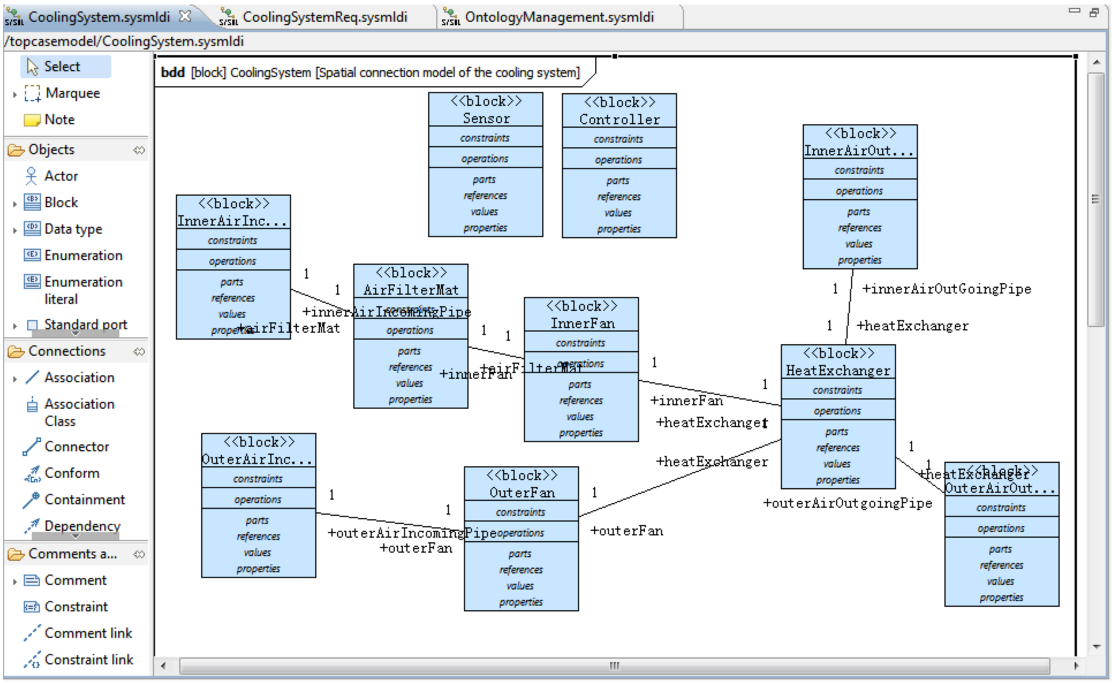


Figure 6-6 Spatial connection modelling

When the system models are built in Topcased, the user needs to go back to the change management system and select a change case which has been created before. Then the user needs to select and upload the functional requirement model file and the physical structure model file (as shown in Figure 6-7). These two files are then automatically converted into a matrix, which is shown in Figure 6-8.

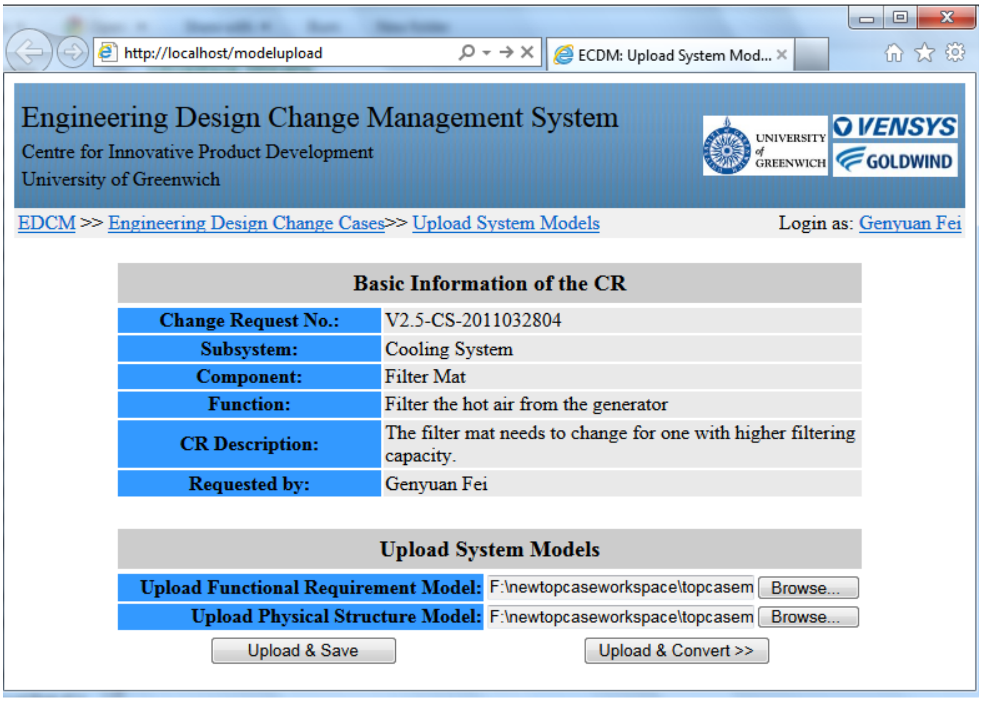


Figure 6-7 System models uploading and transformation

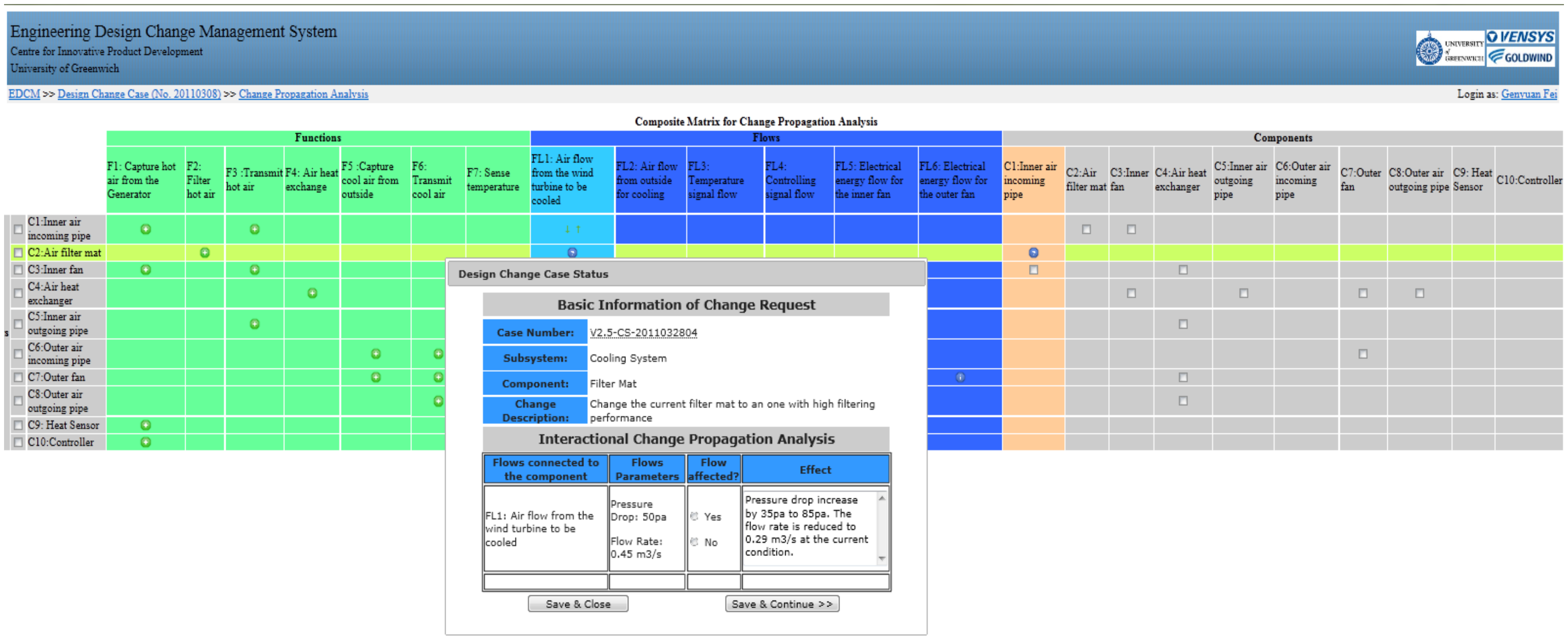


Figure 6-8 The matrix for change propagation analysis

The matrix is in the form of a very long table due to the presentation technology. As depicted in Figure 6-8, the grey column on the far left is a list of components of the subsystem where a design change occurs. The second row of the green zone represents the functional requirements of the subsystem. A marked cell in the matrix section of the green zone represents a component that is involved in the realisation of a functional requirement. The second row of the blue zone represents the flows that connect those components. A marked cell in the rest of the blue zone means that flow goes through the corresponding component. The second row of the grey zone on the right side represents the component. A marked cell in the rest of the grey zone means there are two components spatially connected. The composite matrix at this point has nothing to do with any design change cases, which is just automatically converted from system models.

When the matrix is ready, the component which a change case is associated with is selected for change propagation analysis. Based on the pre-defined relationships in the composite matrix, a popup window is opened, within which the basic information of the change order is shown and also flows that go through the component are listed for analysis. Following the flows, other components that these flows go through are listed as well. Following the components are the functional requirements that the components serve. If the flows going through the original component change because of the change to the original component, the knock-on effects need to be analysed by following this route: the original component → connected flows → connected components → related functional requirements. If there are functional requirement that cannot be satisfied because of the change, it is identified that there is a design conflict that happens to the component in that route right before the functional requirement.

After the analysis of the component-flow-component-functional requirement route, it is also necessary to analyse the component-component-functional requirement route, which means the change to the original component may also cause change to the spatially connected components. If the changes to those components lead to dissatisfactions of their related functional requirements, it is identified that there are design conflicts between the two components.

Figure 6-9 shows the interface of the system for design conflict formalisation with pre-defined ontology. The design conflict is decomposed into function, input flow, output flow and component. The characteristics of the component and the flows that are considered as important factors are also formalised with predefined ontology. For example, in the above interface, the function of this component is ‘heat exchanging’. Clicking the textbox in the knowledge metadata column, a popup window appears which presents the ontology definition. Engineers can choose the appropriate definition to tag (see Figure 6-10).

After selecting concepts for each part of the decomposed design conflict, the form needs to be submitted for knowledge reasoning. The knowledge system will then retrieve the most semantically similar design cases as reference solutions and the engineer can choose from the prioritised list of reference solutions.

Figure 6-9 Design Conflict Formalisation

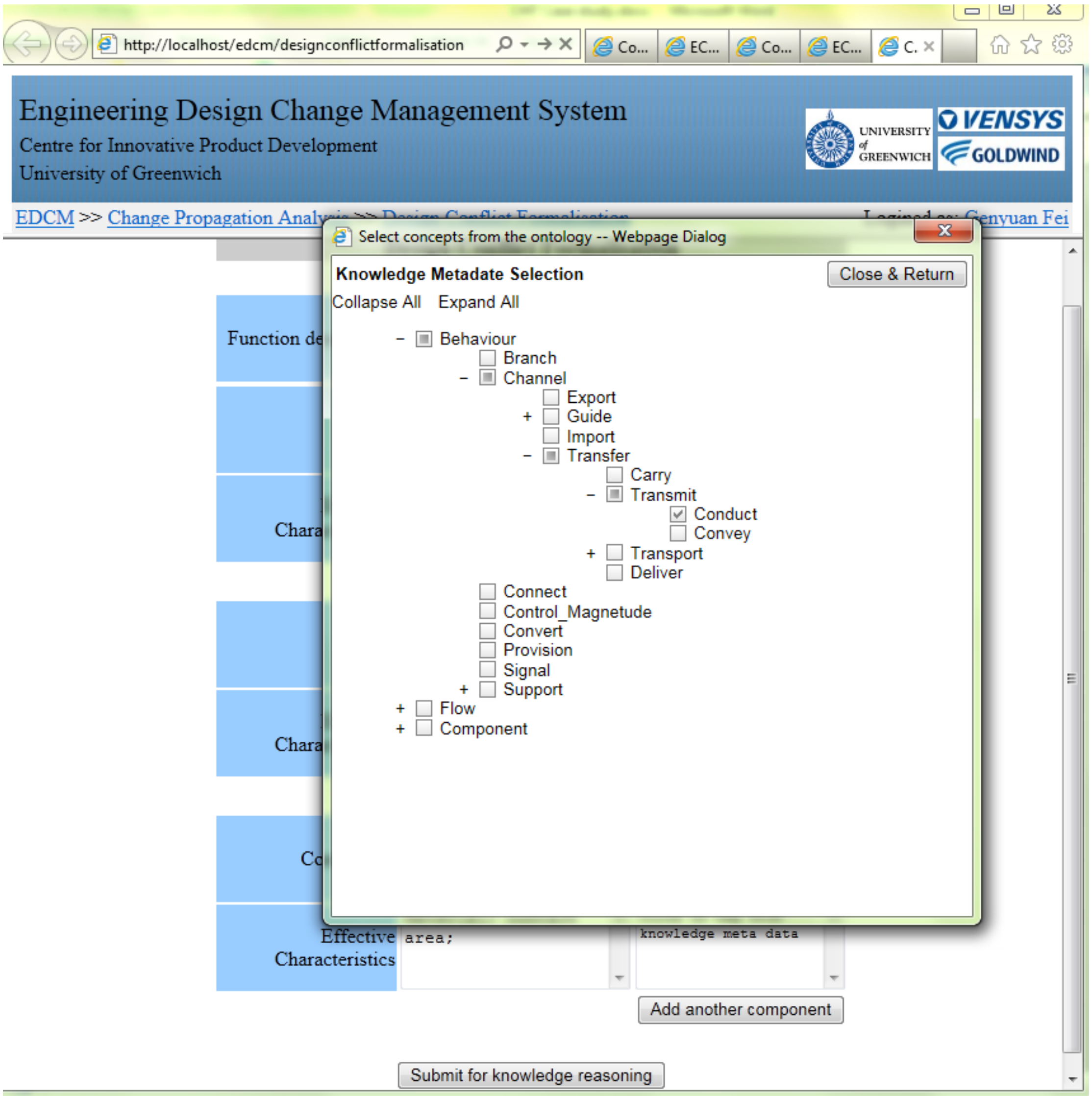


Figure 6-10 Knowledge metadata tagging

6.3 System Validation

In this section, the developed system is assessed by engineers from the investigated company and one of their collaborators. The aim of system validation is to *examine the system from potential users' perspectives and identify the shortcomings for future improvement*. The system validation process is composed of six stages as depicted in Figure 6-11.

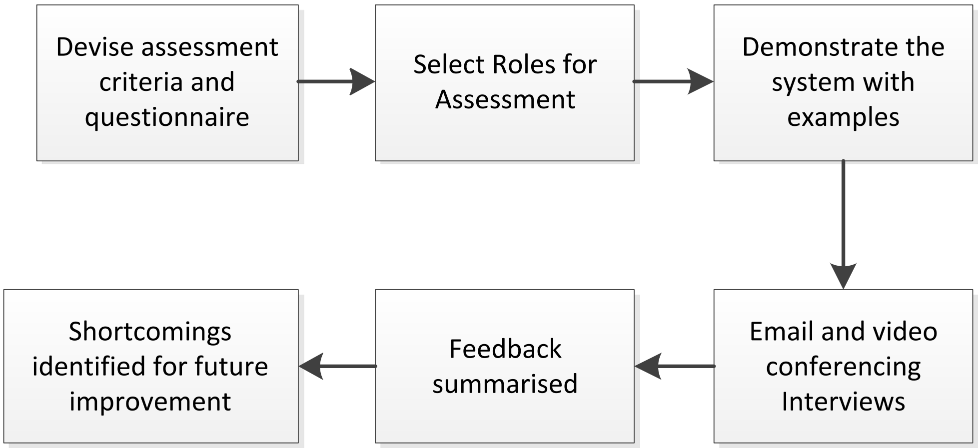


Figure 6-11 The Process of System Validation (By Author)

In the first stage, a questionnaire with validation criteria is devised. The questionnaire mainly covers three top level criteria, i.e., functional coverage, tool usefulness and usability. The overall system and each of its modules will be assessed with these three criteria. Functional coverage as one of the criteria is used to assess the system and its modules to see whether they have covered sufficient functional operations that are used in engineering design change management. Another criterion, tool usefulness, is used to examine whether developed modules and operation processes of the system works competently for the target functions which the modules and processes are initially designed for. The criterion, usability, is focused on the user experience. Even if the intended functions are implemented and work competently in the system, a successful software tool is also required to be easy to use and user friendly, which is one of the important requirements indicated by interviewees from the initial investigation of this research. Bad user experience could very likely lead to failure of the implementation of a software tool.

In the second stage, potential users are selected and asked to fill the questionnaire and give feedback. There are three engineers, two from the investigated company and one from a collaborating company, who took part in the system validation. One engineer from the investigated company is the manager of the new product line. The other one from that company is an engineer who is in charge of a subsystem of the new product.

The engineer from the collaborating company is a liaison engineer between the investigated company and his own company regarding the manufacturing affairs of the new product. The collaborating company is one of the licensees of the investigated company. Therefore, selection of engineers attending the system validation covers the project management, general engineer and collaborative partner. Moreover, the collaborating partner in this case is also an engineer representing the manufacturing phase of product development.

In the third stage, the system is demonstrated with an actual example from the investigated company. Main parts of the demonstration can be found from section 6.2 (page 125), which uses the same example. At the beginning of the demonstration, the developed system is introduced, including the business area (engineering design change management in this case), modules of the system and their main functionalities, and also the process of using the system. Then an engineering design change example is demonstrated from the change request creation to change solution found and evaluated.

In the fourth stage, the questionnaire is answered by participants and interviews are carried out. Due to geographic restriction, internet communication tools are used to carry out the questionnaire and interviews. The questionnaire was sent out and answers are received by emails. Further email communications was also made for further queries regarding some parts of the questionnaire and comments which were not clarified. Two video conferences were also held with the ordinary engineer from the investigated company and the liaison engineer from the collaborative company to get comments and suggestions on the system.

In the fifth stage, feedback from the interviewees is summarised. The feedback is categorised into two main parts, i.e., feedback to the system as a whole and feedback to each module of the system. In each category, the feedback is divided into four parts, i.e. comments on functional coverage, usefulness and usability, and suggestions for further improvement.

In the sixth stage, shortcomings of the proposed methodology and the software system have been identified based on the feedback and suggestions from potential users. Future improvement of the methodology and the system will target these identified shortcomings.

6.4 Findings from the Feedback

As described above, questionnaires are used in this project to interview potential users and collect feedback from them after demonstrating the system with examples. The interviews were carried out in a remote way using internet technologies, including emails and video conferences. The interviewees were asked each question listed in the questionnaires and they provided comments from their experience and impressions of the system. Their answers and feedback are summarised in the remaining parts of this section.

The questionnaires are composed of two main sections. In the first section, questions for each module of the system have been devised. In the second section, general questions regarding the impressions of the whole system have been devised as final verdicts of the system validation. Answers and comments for each question are listed following the question (interviewee 1 who is the manager of the new product line, interviewee 2 who is the ordinary engineer from the investigated company and interviewee 3 is the liaison engineering from the collaborating company). The questionnaires are enclosed in Appendix I.

Based on the analysis of feedback and comments from the questionnaires and further communications, some findings from the potential users are summarised. The findings are summarised in four aspects: the functional coverage, the effectiveness of the methods, the efficiency of the methods, and user experience of the tools. Feedback and comments on each module have been summarised in the following sections. Especially, the shortcomings indicated by interviewees and their suggestions are discussed.

6.4.1 The System Modelling Tool

The system modelling tool is able to present the functional requirements and the physical interactional relationships between components. It also successfully builds the mapping relationships between the functional requirements and the components. It can help engineers understand the product design in an intuitive way. It has also been suggested that the design constraints should be considered in the models since they also play an important part while analysing engineering design changes. The model used to present spatial relationships is not considered as sufficient. An interviewee suggested that CAD models which are widely used to present the spatial relationships between components should be integrated with the system modelling tool.

The interviewees also thought the system modelling is effective for simulating the product design and providing sufficient information for design change analysis. Users can benefit from the tool with better understanding of the design. However, one interviewee indicated that it might be useful to add the interdependent relationships between functional requirements in the functional model. In this methodology, the interdependent relationships are not actually considered. The author argues that although the functional interdependency does exist between the functional requirements, this type of relationships is actually materialised by physical components. So as long as each component can satisfy its corresponding functional requirements without causing any negative effects on operations of other components, the interdependent relationships of functional requirements do not have to be considered. Another interviewee also indicated that the symbols used in the modelling method are not easy to understand. As a standard system modelling language, SysML™ does need some learning and training to completely understand the symbolic meaning of each element.

The modelling tool, Topcased, used in this modelling method is considered as easy to use and learn by interviewees with more design experience, but the interviewee from manufacturing felt the tool was hard to use and learn. The modelling tool is actually derived from software design tool UML. So engineers who are familiar with the software design tool and have experience using software may find it easy while engineers with more manufacturing, rather than software, experience may feel the tool is a little difficult to use. Therefore, in order to successfully implement the tool in industry, it is necessary to carry out some training.

6.4.2 Matrix based Change Propagation Analysis

The matrix is considered as providing a sufficient functional coverage and a novel way of conducting change propagation analysis. By using this tool, the function-component relationships, interactional relationships and spatial relationships between components are clearly presented. As it is transformed from the system models, one of the interviewees has mentioned that the design constraints should be considered in the matrix. Another interviewee also suggested considering the manufacturing factors while analysing change propagation in this matrix. In this research, since this system is developed for change analysis for product design, the manufacturing factors were excluded from the functionality of the target system.

The interviewees also appreciated the sufficient information provided in the composite matrix while analysing change propagation. It provides a very integral and intuitive way to display the connections between each design elements. It helps engineers better understand the design. However, one interviewee indicated that a changed flow might pass one component without causing any negative effects while it may cause negative effects on the next component. He recognised that the composite model was able to capture this type of situation but it was not clearly addressed in the system demonstration.

Besides the effectiveness of analysing change propagation of complex engineering product, the efficiency also becomes a concern of potential users. The web-based technology used in constructing the composite matrix has presented an interactive way to provide a lot of information and makes it easier for users to understand the relationships. Interviewees indicated that if the complexity of an engineering product is significant the current design of the composite matrix may make change propagation analysis less efficient and eventually very hard to use.

In terms of user experience, due to the light-weighed and intuitive web presentation technology, the interviewees agree that the composite matrix successfully provides an easy-to-use and easy-to-learn tool for engineering change propagation analysis.

6.4.3 Knowledge Management and Knowledge Use for Design Conflict Solving

The knowledge management module is considered as a good mechanism for design knowledge capturing and knowledge retrieval. It is considered as providing a sufficient functional coverage for knowledge management in terms of use for design conflict solving. Some suggestions were put forward in order to improve the knowledge management module. One interviewee indicated that it might be better if there is an integration interface available, which can obtain design cases directly from existing product information management system, e.g., product data management system (PDM). Another interviewee suggested that it might be better if the engineers are allowed to build the ontology and a tool can be provided for assessing the ontology built by them. Therefore, a community can be formed in the company and everyone can contribute to the knowledge building process. But there is a concern by the author that a community-managed ontology building mechanism could lead to loss of control of the consistency maintenance.

Besides, the interviewees appreciate the idea of ontologically formalising problems or requirements, which design solutions are proposed for, to index the design cases. The method is effective to capture design knowledge from engineers' everyday work and enable engineers to contribute to the knowledge repository gradually. The knowledge entries that are formalised with the pre-defined ontology are also easier to be retrieved by comparing the semantic similarities with the target design conflicts to be solved. They also agree that the ontology-formalised knowledge entries help engineers find existing solutions which are proposed by other engineers. It is also easier to understand solutions without going through large amount of design documents.

But they also indicated that the efficiency may be questionable to initialise the knowledge repository since it requires a lot of work to transform the current design cases, which are stored in paper-based technical documents and electronic document management systems, into knowledge entries. Besides, one interviewee came up with a doubt whether ordinary engineers without experience of knowledge management can correctly choose right concepts from the pre-defined ontology to formalise the problems or requirements.

The user experience of the knowledge management module seems satisfactory. Again, due to the web-based presentation technology, users found it was easy to learn how to use the system. There is no heavy training work needed to use such a system.

6.4.4 Change Solution Evaluation

The functional coverage of the change solution evaluation module is found to be sufficient. However, one interviewee raised a concern that the criteria selected for evaluating the solutions could be inadequate for some specific projects. Although he did not come up with any suggestions about which other criteria should be considered in the evaluation process, it might be a good idea to improve the module with an open mechanism which allows engineers to configure the criteria for certain change solutions according to the nature of a specific project.

The web service based integration mechanism designed in the solution evaluation module was considered as open and flexible, which makes future integration with other enterprise systems easier. Although one interviewee questioned the security of using such an open integration method, it is not a major concern in this research. This is because, on one hand, the security measures for integration are not in the scope of this research. On the other hand, there are mature security measures for web service based integration mechanism and these are easy to implement.

In terms of user experience, the interviewees agreed that it is easy to use and learn with this module. The integration part of the module may be a little technical but it is not a concern for end users.

6.4.5 Discussion

Not only can the proposed methodology be used in the engineering design area to analyse engineering design changes, some parts of it can be also used separately to help engineering design. For example, the composite matrix is built based on three relationships, i.e. the mapping relationship between functional requirements and the physical components, the international relationship between components and also the spatial connection between components. So it can be used to help engineering designers analyse coupling relationships between components for design improvement. The component, which has involved in realisations of more functional requirements and has more physical interactions and connections with others, means it gets more coupled with others. Therefore if there is a problem happens to this component, the potential cost to solve the problem may be higher than others. So designers can focus on this component to carry out some improvements to reduce its coupling degree.

For another example, the knowledge base is built based on the basic functional model which is composed of input flow, output flow, functional description and component description. So it can also be used to solve design problems that are not only just for engineering change but also for general engineering design since many design problems can be decomposed into these basic elements.

However, beyond the design domain, the usefulness of this methodology is very limited. On one hand, in the very early design stage, the so-called front end stage, the components are not considered and therefore the composite matrix will not be able to build. And also design problems at this stage involve more about customer requirements analysis, which makes the knowledge base not suitable to use. On the other hand, if it's at the detail design stage, engineers are focused more on parametric design and design for manufacturing which normally does not need to consider too much about functional requirements and interactional relationships. So the methodology is very limited to use at this stage as well.

6.5 Summary of the Validation

The system validation is carried out in form of interviews and questionnaires. Three potential users have participated in the system validation. Questionnaires have been designed for collecting feedback and comments from them. The questionnaires main cover four modules of the system, i.e., the system modelling module, the change propagation analysis module, the knowledge management module and the change solution evaluation module. Questions for each module basically cover four aspects which include functional coverage, effectiveness, efficiency and user experience. As a whole, the interviewees agreed that this system for engineering design change analysis provides a sufficient functional coverage. It has high effectiveness and fair efficiency in analysing engineering design change. The web-based technology provides an easy and intuitive way to carry out the work.

Questions and concerns were raised by interviewees. Some of them were raised because of the misunderstanding of the development intentions of this system. Some are because of the backgrounds of different engineers. Indeed, some questions and concerns are regarded as important shortcomings of the system and thus are considered for future improvement. The interviewees also provided some valuable suggestions which will be seriously considered as further work of this research.

CHAPTER 7 DISCUSSION, CONCLUSIONS AND FURTHER WORK

In the chapter, the conclusions of the research project are presented. Further work of this research is also indicated according to the feedback and comments from potential users.

7.1 Discussion

Based on research gaps found from a review of state-of-the-art research work and industrial needs identified from investigation in collaborative companies, a model-driven and knowledge-based methodology is developed in this project to analyse engineering change in the product design phase, resolve design conflicts arising from engineering change and evaluate change solutions.

From a broad review of research literature in the early stage of this project, it is found that current methods proposed for engineering change management do not take sufficient consideration of the influence between the functional domain and the structural domain of product design. Most research work in engineering change management focuses on document consistency, product data dependency, or physical component dependency. The consistency between the functional requirements and their corresponding physical structures are not systematically considered in engineering design change analysis. Also as a critical part of engineering design change management, current research on change propagation analysis is mainly based on estimating and predicting methods, it is found that propagated changes largely depend on solutions of previous causal changes. Since change solutions may vary from case to case, estimating and predicting methods, which depends on predefined physical dependencies, may not be effective for analysing change propagations. Consequently, following change impact analyses based on these methods are not considered as reliable. These research gaps identified in literature review in the early stage are focused on and determined as domain of interest of this research project.

An industrial investigation has been carried out in the collaborative company, which is a small and medium enterprise (SME) and has a successful business in design and development of wind turbines. It is a typical R&D focused company and has broad international collaborations. Several research methods have been adopted in this investigation in order to spend less resources of the company and obtain as much information as possible. It is found that engineering design changes cause serious delay of new product development and cost a lot to improve and maintain the old product line. There is a lack of systematic method to analyse engineering design change and resolve emerging design conflicts. It is identified that design knowledge, which is critical to the company's business, is not systematically managed. Although members of staff recognise the value and importance of their design knowledge from previous design cases, there is a lack of method to reuse it to solve current design problems. It is also found that there is a lack of systematic method to evaluate design change solutions which currently depends on individual experience. With the identified industrial needs, further literature review was carried out. It is found that there is a lack of research addressing knowledge management and reuse in engineering design change management and design conflict solving.

In order to bridge the research gaps and meet the industrial needs, a model-driven and knowledge-based methodology has been proposed to analyse engineering design change, solve design conflicts and evaluate change solutions. To clarify the dependency and interactions between elements of a product design, a system engineering modelling language is used to model the functional requirements, physical connections and interactions between components. The system engineering models are then transformed into a composite matrix model which is used to analyse and trace design change propagations and identify design conflicts emerging from change propagations. A knowledge-based method is then developed to help solve these design conflicts. Previous design cases are formalised and reused in this method by applying predefined domain ontology. An ontology rating approach and a semantic similarity comparison approach are developed in order to help retrieve similar design cases according to the semantic meanings of target design conflicts. An evaluation method is also proposed which can evaluate change solutions with consideration of factors of project success.

A prototype software system is developed in this project to implement the proposed methodology. The necessity of developing a software system for such a methodology is discussed. Evaluation and selection of tools for the prototype system is also discussed. The architecture of the whole system is presented together with tools and technologies that it employs. In-depth explanation is provided regarding how the system works. Also some key techniques developed in the system are presented and discussed.

A case study was carried out to apply the developed prototype system with a real industrial example. The functions implemented in the software system was tested and demonstrated to potential users. In order to validate the system with potential users, questionnaires were developed and individual interviews were carried out. During the system validation process, internet-based technologies were used for remote communication, which include Email and web conferencing. The questionnaires and interviews mainly covered three aspects, i.e., functional coverage, effectiveness of the methods, efficiency of the methods, and user experience of the tools. Each module of the system was validated individually and then the final verdict of the whole system was made. Feedback and comments received from the potential users were summarised and analysed. It is found that the system as a whole covers the main needs for engineering design change analysis. It is also agreed by the potential users that the system provides effective, acceptably efficient and easy to use approaches to dealing with engineering design change. However, comments and suggestions are also received for functionality enhancement, usability improvement and shortcomings correction. These comments and suggestions will be used as directions for further work of this project.

7.2 Conclusions

The main achievements of this research project are briefly summarised below corresponding to the initial objectives:

Objective 1: To investigate previous and current research and development work in engineering design change management and identify research gaps and industrial requirements;

Corresponding achievements:

- Reviewed state-of-the-art research work in engineering change management in product development and identified research gaps in chapter 2;

- Carried out an industrial investigation and identified industrial needs in engineering change management and design knowledge use in chapter 3;

Objective 2: To propose a framework for design change management in and between functional and physical domains of product design;

Corresponding achievements:

- Proposed a framework and a working process for engineering design change analysis in chapter 4 section 4.2 and section 4.3;

Objective 3: To propose a model driven method to analyse change propagation and identify design conflicts arising from change propagation;

Corresponding achievements:

- Proposed a method using SysML™ and design matrix to model the relationships in and between functional requirements and physical components in chapter 4 section 4.4.1;

- Proposed an approach to analysing design change propagation and identify design conflicts based on the composite matrix in chapter 4 section 4.4.2;

Objective 4: To propose a knowledge-based method to solve design conflicts by reusing previous design cases;

Corresponding achievements:

- Developed a knowledge base by using ontology technology and Case Based Reasoning (CBR) technology to reuse and formalise previous design cases (see in chapter 4 section 4.5);

- Developed a knowledge reasoning approach to find semantically similar solutions to help solve design conflicts (see in chapter 4 section 4.5);

Objective 5: To propose a method to evaluate and prioritise change propagation paths in terms of impacts on project success;

Corresponding achievements:

- Proposed a method to evaluate candidate solutions with consideration of development time, development cost, and development risk which are considered as important factors of project success (see in chapter 4 section 4.6);

Objective 6: To develop a prototype system for demonstrating and evaluating the proposed methodology;

Corresponding achievements:

- Developed a computer aided system using open-source tools and web-based technologies to implement the proposed methodology (see in chapter 5);

Objective 7: To develop a case study for validating the whole research project using industrial applications and interviewing users;

Corresponding achievements:

- Carried out a case study with an industrial example and evaluated the tool by interviewing potential users. Findings from the case study have been discussed (see in chapter 6).

7.3 Limitations and Further Work

7.3.1 Limitations of the methodology

Although the proposed methodology and software tool have been considered as useful for engineering design change analysis, there are limitations of the methodology and the system according to feedback and comments received from potential users.

Firstly, industrial investigation and case study are limited. The industrial investigation of this research was primarily carried out in one manufacturing company where two product lines are ongoing. Although the investigated company was selected to meet the requirements of industrial investigation of this research, the investigated business areas in manufacturing companies and outcomes from the case study are still relatively limited. A broader investigation in the future with more industrial requirements found could be very useful to improve the methodology. Further case studies with more products can also be very helpful to find more limitations of the methodology and the system.

Secondly, the model-driven change propagation analysis method is limited since it does not consider the design constraints. Within the proposed methodology, the functional requirement domain and the physical structure domain have been considered to analyse engineering design change. However, as suggested by the potential users in system validation, it may be more useful to take design constraints into consideration as well. Design constraints may not add functionalities to a product design, but they may be potentially violated by a change to a component, which may make the change solution invalid or require further change to address the violation. This situation is not fully considered in the proposed methodology.

Thirdly, the efficiency of knowledge base building is not high. The knowledge base is supported by a lot of design cases and the effort to formalise these design cases. In the current development of the methodology, it needs a lot of time from engineers to obtain the information of design cases and formalise them using the proposed approach. Therefore, like many other knowledge management systems, the efficiency of depositing the previous design information and formalising them as knowledge entries will be a challenge for the company and make the implementation of the system very expensive. Some automated methods need to be put in place to reduce the work of knowledge base building.

Finally, the method for evaluating candidate solutions has a lack of flexibility. In this research, there are actually four factors are taken into consideration in candidate solution evaluation, i.e. development time, development cost, development risk and functional consistency. These four factors are intended to represent the overall performance of product development project. However, as suggested by one of the interviewees in case study, there may be other factors that are also very important to some particular projects in some circumstances. Therefore, this evaluation method should not be bound only to the pre-specified criteria. Flexibility needs to be considered to allow users to define criteria that are more suitable for their projects.

7.3.2 Further work

Further work of this research is planned based on the identified limitations of the methodology and the some feedback and comments given by some interviewees who thought there were some spaces for improvement according to their experience.

Firstly, the modelling-driven approach to engineering design propagation analysis needs to be improved with consideration of design constraints. Further investigation and exploration of how design constraints may influence engineering design change analysis need to be carried out in future work. The outcome of the investigation and exploration may help improve the current methodology to accommodate that need.

Secondly, the web-based presentation of the composite matrix needs to be improved. It is found in the system validation that the composite matrix, which is converted from system engineering models of a product design for change propagation analysis, may be difficult to use if there is a complex product with a lot of functional requirements and components. The presentation approach for the composite matrix may need improvement with interactive web technologies on the client side (the internet browser), for example the AJAX (Asynchronous JavaScript and XML) technique may be useful to break down the matrix with flexible views.

Thirdly, the knowledge management module needs further improvements. According to a concern brought up by an interviewee, improvement to the knowledge management module of the system may be needed. It is suggested that an integration approach needs to be in place to import engineering design cases that are stored in their current enterprise systems, e.g. product data management system, content management system, into the knowledge management module to reduce the workload. Also as suggested, the ontology management in the knowledge management module needs to be improved with a flexible and controlled method for ontology definition. Therefore, ordinary engineering designers can take part in improving the domain ontology definition. An instruction or guideline integrated in the knowledge management module will also be developed to help designer formalise design cases to be knowledge entries.

Finally, a more flexible approach for change solution evaluation is needed. The change solution evaluation method in this system uses a web service based approach to integrate with other enterprise application system so that information regarding development time, development cost and development risk of a component can be obtained. However, as mentioned in the section 7.3.1, the pre-defined criteria may be not able to meet the requirements of various projects since different projects may need to consider more and different criteria when the solution is evaluated. Further work for this part of the system is to increase the flexibility of the evaluation approach so that users will be able to configure the evaluation criteria in the interest of their specific product development projects.

REFERENCES

Aamodt, A. and Plaza, E. 1994. Case-based reasoning: foundational issue, methodological variation, and system approaches. *Artificial Intelligence Communications*, 7(1), 39-59.

Alter, K., 2009. *Concise Oxford English Dictionary 2009.* London: Oxford University Press.

Altshuller, G. S., 1996. *And suddenly the inventor appeared : TRIZ, the theory of inventive problem solving.* 2nd ed. Worcester, Mass.: Technical Innovation Center.

Ariyo, O. O., Eckert, C. M. and Clarkson, P. J. 2009. Challenges in identifying the knock-on effects of engineering change. *International Journal of Design Engineering*, 2, 414-431.

Balogun, J. and Jenkins, M. 2003. Re-conceiving Change Management:: A Knowledge-based Perspective. *European Management Journal*, 21(2), 247-257.

Black, R., 2010. *China steams ahead on clean energy* [online]. London: BBC. Available from: http://news.bbc.co.uk/1/hi/8587319.stm [Accessed 5th Feburary 2011].

Boznak, R. G. and Decker, A. K., 1993. *Competitive product development : a quality approach to succeeding in the '90s and beyond.* Milwaukee, Wis.: ASQC Quality Press ; Homewood, Ill. : Business One Irwin.

Bradfield, D. J. and Gao, J. X. 2007. A methodology to facilitate knowledge sharing in the new product development process. *International Journal of Production Research*, 45(7), 1489-1504.

Brandt, S. C., Morbach, J., Miatidis, M., Theißen, M., Jarke, M. and Marquardt, W. 2008. An ontology-based approach to knowledge management in design processes. *Computers & Chemical Engineering*, 32(1-2), 320-342.

Brewer, H., Automated tool design: age forming tool for aerospace panels. ed. *International ICAD Users Group Conference*, February 1996.

British Department for Business Innovation & Skills, 2010. *Manufacturing in the UK: Supplementary analysis* [online]. British Department for Business Innovation & Skills. Available from: http://www.bis.gov.uk/assets/biscore/business-sectors/docs/m/10-1334-manufacturing-in-the-uk-supplementary-analysis.pdf [Accessed 9th Feburary 2011].

Brown, J., 2006. *Managing Product Relationships: Enabling Iteration and Innovation in Design.* Boston: AberdeenGroup.

Buckley, F. J., 1996. *Implementing configuration management : hardware, software, and firmware.* 2nd ed. ed. Los Alamitos, Calif.: Computer Society Press.

Chang, A. S.-T., Shih, J. S. and Choo, Y. S. 2010. Reasons and costs for design change during production. *Journal of Engineering Design*.

Chang, X., Sahin, A. and Terpenny, J. 2008. An ontology-based support for product conceptual design. *Robotics and Computer-Integrated Manufacturing,* 24(6), 755-762.

Chapman, C. B. and Pinfold, M. 2001. The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure. *Advances in Engineering Software,* 32(12), 903-912.

Chen, G., Ma, Y.-S., Thimm, G. and Tang, S.-H. 2004. Unified feature modeling scheme for the integration of CAD and CAx. *Computer-Aided Design & Applications,* 1(1-4), 595-602.

Chen, Y.-M., Shir, W.-S. and Shen, C.-Y. 2002. Distributed engineering change management for allied concurrent engineering. *International Journal of Computer Integrated Manufacturing,* 15(2), 127 - 151.

Clarkson, J. and Eckert, C., 2005. *Design process improvement: a review of current practice.* London: Springer.

Clarkson, P. J., Simons, C. and Eckert, C. 2004. Predicting Change Propagation in Complex Design. *Journal of Mechanical Design,* 126(5), 788-797.

Conrad, J., Deubel, T., Köhler, C., Wanke, S. and Weber, C., Change impact and risk analysis (CIRA) –Combining the CPM/PDD theory and FMEA-methodology for an improved engineering change management. ed. *International conference on engineering design, ICED'07,* 28-31 August 2007 Paris, France.

Cooper, R. G. and Edgett, S. J. 2006. Stage-gate and the critical success factors for new product development. *BPT Trends,* 7, 1-6.

Cooper, R. G. and Kleinschmidt, E. J. 1995. Benchmarking the Firms Critical Success Factors in New Product Development. *Journal of Product Innovation Management,* 12, 374-391.

Corcho, O., Fernández-López, M. and Gómez-Pérez, A. 2003. Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering,* 46(1), 41-64.

Crnkovic, I., Asklund, U. and Dahlqvist, A. P., 2003. *Implementing and integrating product data management and software configuration management.* Boston ; London: Artech House.

Darlington, M. J. and Culley, S. J. 2008. Investigating ontology development for engineering design support. *Advanced Engineering Informatics*, 22(1), 112-134.

Dec, Fujitsu, Ibm, One, M., Metaphase, Sdrc and Sherpa, 1998. Product Data Management Enablers RFP. *Manufacturing Domain Task Force Docs* [online], mfg/96-08-01. Available from: ftp://ftp.cs.rutgers.edu/pub/borgida/pdm-joint-98-01-01.pdf [Accessed 13th Feburary 2010].

Dennis, A., Wixom, B. H. and Tegarden, D. P., 2002. *Systems analysis and design : an object-oriented approach with UML.* New York ; Chichester: Wiley.

Denny, M., 2004. *Ontology Tools Survey, Revisited* [online]. Available from: http://www.xml.com/pub/a/2004/07/14/onto.html [Accessed 21st April 2011].

Dieter, G. E. and Schmidt, L. C., 2009. *Engineering design.* International ed., 4th ed. ed. London: McGraw-Hill.

Do, N., Choi, I. J. and Song, M. 2008. Propagation of engineering changes to multiple product data views using history of product structure changes. *International Journal of Computer Integrated Manufacturing*, 21(1), 19 - 32.

Duineveld, A. J., Stoter, R., Weiden, M. R., Kenepa, B. and Benjamins, V. R. 2000. WonderTools? A comparative study of ontological engineering tools. *International Journal of Human-Computer Studies*, 52(6), 1111-1133.

Eckert, C., Clarkson, P. J. and Zanker, W. 2004. Change and customisation in complex engineering domains. *Research in Engineering Design*, 15(1), 1-21.

Eckert, C. M., Keller, R., Earl, C. and Clarkson, P. J. 2006. Supporting change processes in design: Complexity, prediction and reliability. *Reliability Engineering & System Safety*, 91(12), 1521-1534.

Ehrlenspiel, K., Kiewert, A., Lindemann, U. and Hundal, M. S., 2007. *Cost-Efficient Design.* London: Springer

Fey, V. and Rivin, E. I., 2005. *Innovation on demand: new product development using TRIZ.* Cambridge: Cambridge University Press.

Fitzsimmons, J. A., Kouvelis, P. and Mallick, D. N. 1991. Design strategy and its interface with manufacturing and marketing: A conceptual framework. *Journal of Operations Management*, 10(3), 398-415.

Folkmanis, J. and Giang, N. K., 2010. *Vietnam Tycoon Lures China Companies With Cheap Labor* [online]. Bloomberg. Available from: http://www.bloomberg.com/apps/news?pid=newsarchive&sid=a121aZu_HTz0 [Accessed 10th Feburary 2011].

Gerwin, D. 1982. Do's and don'ts of computerized manufacturing. *HARVARD BUSINESS REVIEW*, 60, 107-116.

Giffin, M., De Weck, O., Bounova, G., Keller, R., Eckert, C. and Clarkson, P. J. 2009. Change Propagation Analysis in Complex Technical Systems. *Journal of Mechanical Design*, 131(8), 081001-081014.

Goel, A., Bhatta, S. and Stroulia, E., 1997. KRITIK: an early case-based design system. *In:* Maher, M. and Pu, P. eds. *Issues and applications of case-based reasoning in design.* Mahwah, NJ: Erlbaum, 87-132.

Goel, A. K. and Craw, S. 2005. Design, innovation and case-based reasoning. *Knowl. Eng. Rev.*, 20(3), 271-276.

Gruber, T. R. 1993. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2), 199-220.

Han, Y.-H. and Lee, K. 2006. A case-based framework for reuse of previous design concepts in conceptual synthesis of mechanisms. *Computers in Industry,* 57(4), 305-318.

Heinz, A., 1996. 777 Rule Based Design - Integrated Fuselage System. *International ICAD Users Group Conference.*

Helms, R. W., 2002. *Product Data Management as enabler for Concurrent Engineering.* University of Technology.

Hirtz, J., Stone, R., Mcadams, D., Szykman, S. and Wood, K. 2002. A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts. *Research in Engineering Design,* 13(2), 65-82.

Huang, G. Q. and Mak, K. L. 1999. Current practices of engineering change management in UK manufacturing industries. *International Journal of Operations & Production Management,* 19(1), 21-37.

Huang, G. Q., Yee, W. Y. and Mak, K. L. 2001. Development of a web-based system for engineering change management. *Robotics and Computer-Integrated Manufacturing,* 17, 255-267.

Huang, G. Q., Yee, W. Y. and Mak, K. L. 2003. Current practice of engineering change management in Hong Kong manufacturing industries. *Journal of Materials Processing Technology,* 139(1-3), 481-487.

Janthong, N., Brissaud, D. and Butdee, S. 2010. Combining axiomatic design and case-based reasoning in an innovative design methodology of mechatronics products. *CIRP Journal of Manufacturing Science and Technology,* 2(4), 226-239.

Jarratt, T. a. W., Eckert, C. M., Weeks, R. and Clarkson, P. J., Environmental legislation as a driver of design. ed. *The 14th International Conference on Engineering Design (ICED'03),* 19-21 August 2003 Stockholm, Sweden.

Kajko-Mattsson, M., 1999. Maintenance at ABB (II): Change Execution Processes. *Proceedings of the IEEE International Conference on Software Maintenance.* IEEE Computer Society, 307.

Kalpic, B. and Bernus, P. 2002. Business process modelling in industry--the powerful tool in enterprise management. *Computers in Industry,* 47(3), 299-318.

Keese, D. A., Seepersad, C. C. and Wood, K. L. 2009. Product flexibility measurement with enhanced Change Modes and Effects Analysis CMEA. *International Journal of Mass Customisation,* 3, 115-145.

Keller, A., 2005. Automating the Change Management Process with Electronic Contracts. *Proceedings of the Seventh IEEE International Conference on E-Commerce Technology Workshops.* IEEE Computer Society, 99-108.

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

Kidd, M. W. and Thompson, G. 2000. Engineering design change management. *Integrated Manufacturing Systems,* 11(1), 74-77.

Kim, K.-S. and Han, I. 2001. The cluster-indexing method for case-based reasoning using self-organizing maps and learning vector quantization for bond rating cases. *Expert Systems with Applications,* 21(3), 147-156.

Kim, W. C. and Mauborgne, R. E., 2005. *Blue ocean strategy : how to create uncontested market space and make the competition irrelevant.* Boston, Mass.: Harvard Business School.

Kitamura, Y., Kashiwase, M., Fuse, M. and Mizoguchi, R. 2004. Deployment of an ontological framework of functional design knowledge. *Advanced Engineering Informatics,* 18(2), 115-127.

Kitamura, Y. and Mizoguchi, R., Functional Ontology for Functional Understanding. ed. *Twelfth International Workshop on Qualitative Reasoning*, May 26-29 1998 Cape Cod, USA.

Kitamura, Y. and Mizoguchi, R. 2003. Ontology-based description of functional design knowledge and its use in a functional way server. *Expert Systems with Applications,* 24(2), 153-166.

Kitamura, Y. and Mizoguchi, R. 2004. Ontology-based systematization of functional knowledge. *Journal of Engineering Design,* 15(4), 327-351.

Kitamura, Y., Washio, N., Koji, Y., Sasajima, M., Takafuji, S. and Mizoguchi, R., An Ontology-Based Annotation Framework for Representing the Functionality of Engineering Devices. ed. *ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, September 10-13 2006 Philadelphia, USA.

Kocar, V. and Akgunduz, A. 2010. ADVICE: A virtual environment for Engineering Change Management. *Computers in Industry,* 61(1), 15-28.

Kochan, A. 1999. Jaguar uses knowledge-based tools to reduce model development times. *Assembly Automation*, 19(2), 114-117.

Koh, E. C. Y. and Clarkson, P. J., 2009. A modelling method to manage change propagation. ed. *Proceedings of the 17th International Conference on Engineering Design (ICED'09)*, 24-27 August 2009 Stanford, California, USA.

Köhler, C., Conrad, J., Wanke, S. and Weber, C., 2008 A matrix representation of the CPM/PDD approach as a means for change impact analysis. ed. *The 10th International Design Conference*, May 19 - 22 2008 Dubrovnik, Croatia.

Kolodner, J. L. 1983a. Maintaining organization in a dynamic long-term memory. *Cognitive Science*, 7(4), 243-280.

Kolodner, J. L. 1983b. Reconstructive memory: A computer model. *Cognitive Science*, 7(4), 281-328.

Kolodner, J. L., 1993. *Case-based reasoning.* San Mateo, CA: Morgan Kaufmann Publishers.

Kraft, B. and Nagl, M. 2007. Visual knowledge specification for conceptual design: Definition and tool support. *Advanced Engineering Informatics*, 21(1), 67-83.

Lam, W., 1998. Change Analysis and Management in a Reuse-Oriented Software Development Setting. *Proceedings of the 10th International Conference on Advanced Information Systems Engineering.* Springer-Verlag, 219-236.

Lambrix, P., Habbouche, M. and Pérez, M. 2003. Evaluation of ontology development tools for bioinformatics. *Bioinformatics*, 19(12), 1564-1571.

Lee, H., Ahn, H., Kim, J. and Park, S. 2006. Capturing and reusing knowledge in engineering change management: A case of automobile development. *Information Systems Frontiers*, 8(5), 375-394.

Lee, H., Seol, H., Sung, N., Hong, Y. S. and Park, Y. 2010. An analytic network process approach to measuring design change impacts in modular products. *Journal of Engineering Design*, 21(1), 75-91.

Loch, C. H. and Terwiesch, C. 1999. Accelerating the Process of Engineering Change Orders: Capacity and Congestion Effects. *Journal of Product Innovation Management*, 16(2), 145-159.

Lovett, P. J., Ingram, A. and Bancroft, C. N. 2000. Knowledge-based engineering for SMEs -- a methodology. *Journal of Materials Processing Technology*, 107(1-3), 384-389.

Ma, Y., Chen, G. and Thimm, G. 2008. Change propagation algorithm in a unified feature modeling scheme. *Computers in Industry*, 59(2-3), 110-118.

Mason, P., 2010. *Low wage costs attract investors to Vietnam* [online]. London: BBC. Available from: http://www.bbc.co.uk/news/10344233 [Accessed 10th feburary 2011].

Maull, R., Hughes, D. and Bennett, J. 1992. The role of the bill-of-materials as a CAD/CAPM interface and the key importance of engineering change control. *Comput. Control Eng. J.,* 3(2), 63-70.

Mcintosh, K., 1995. *Engineering data management : a guide to successful implementation.* London: McGraw-Hill.

Mckay, K., Bramall, D., Rogers, B., Chapman, P., Cheung, W. and Maropoulos, P. 2003. Design change impact analysis during early design specification. *International Journal of Computer Integrated Manufacturing,* 16(7), 598 - 604.

Mellows-Facer, A., 2010. *Manufacturing* [online]. House of Commons. Available from: http://www.parliament.uk/briefingpapers/commons/lib/research/briefings/snep-01942.pdf [Accessed 10 Feburary 2011].

Mohan, K., Xu, P., Cao, L. and Ramesh, B. 2008. Improving change management in software development: Integrating traceability and software configuration management. *Decision Support Systems,* 45(4), 922-936.

Mok, C. K., Hua, M. and Wong, S. Y. 2008. A hybrid case-based reasoning CAD system for injection mould design. *International Journal of Production Research,* 46(14), 3783 - 3800.

National Statistics, 2006. *United Kingdom National Accounts: The Blue Book 2006* [online]. Office for National Statistics (ONS). Available from: http://www.statistics.gov.uk/downloads/theme_economy/BlueBook2006.pdf [Accessed 08 Feburary 2011].

Navinchandra, D., 1991. *Exploration and innovation in design : towards a computational model.* New York ; London: Springer-Verlag.

Nichols, K. 1990. Getting Engineering Changes under Control. *Journal of Engineering Design,* 1(1), 5 - 15.

Object Management Group, 2008. *OMG Systems Modeling Language (OMG SysML) V1.1* [online]. Object Management Group. Available from: http://www.omg.org/spec/SysML/1.1/PDF [Accessed 16 May 2009].

Oldham, K., Kneebone, S., Callot, M., Murton, A. and Brimble, R., 1998. MOKA-A Methodology and tools Oriented to Knowledge-based engineering Applications. *In:* Martensson, N., Mackay, R. and Bjorgvinsson, S. eds. *Changing the ways we work : shaping the ICT-solutions for the next century.* Ohmsha: IOS Press, 198 - 207.

Ouertani, M. Z. 2008. Supporting conflict management in collaborative design: An approach to assess engineering change impacts. *Computers in Industry,* 59(9), 882-893.

Ouertani, M. Z. and Gzara, L. 2008. Tracking product specification dependencies in collaborative design for conflict management. *Computer-Aided Design,* 40(7), 828-837.

Pahl, G., Beitz, W. and Wallace, K., 1996. *Engineering design : a systematic approach.* 2nd ed. London: Springer.

Palani Rajan, P. K., Van Wie, M., Campbell, M. I., Wood, K. L. and Otto, K. N. 2005. An empirical foundation for product flexibility. *Design Studies,* 26(4), 405-438.

Park, S. and Bae, D.-H. 2011. An approach to analyzing the software process change impact using process slicing and simulation. *Journal of Systems and Software,* 84(4), 528-543.

Pikosz, P. and Malmqvist, J., 1998. A comparative study of engineering change management in three Swedish enginnering companies. ed. *Design Engineering Technical Conferences and Computers in Engineering Conference*, September 13-16 1998 Atlanta, Georgia.

Pinto, H. S. and Martins, J. P. 2004. Ontologies: How can They be Built? *Knowledge and Information Systems,* 6(4), 441-464.

Pricewaterhousecoopers, 2009. *The future of UK manufacturing: Reports of its death are greatly exaggrated* [online]. PricewaterhouseCoopers. [Accessed 05th Feburary 2011].

Rios, J., Roy, R. and Lopez, A. 2007. Design requirements change and cost impact analysis in airplane structures. *International Journal of Production Economics,* 109(1-2), 65-80.

Saaty, T. L., 1988. *The analytic hierarchy process : planning, priority setting, resource allocation.* 2nd ed. ed.: Thomas Saaty.

Saaty, T. L., 1996. *Decision making with dependence and feedback : the analytic network process.* Pittsburgh, PA: RWS Publications.

Schank, R. C. and Abelson, R. P., 1977. *Scripts, plans, goals and understanding: An inquiry into human knowledge structures.* Oxford, England: Lawrence Erlbaum.

Shiau, J.-Y. and Wee, H. M. 2008. A distributed change control workflow for collaborative design network. *Computers in Industry,* 59(2-3), 119-127.

Stanev, S., Krappe, H., Ola, H. A., Georgoulias, K., Papakostas, N., Chryssolouris, G. and Ovtcharova, J. 2008. Efficient change management for the flexible production of the future. *Journal of Manufacturing Technology Management,* 19(6), 712-726.

Steffens, W., Martinsuo, M. and Artto, K. 2007. Change decisions in product development projects. *International Journal of Project Management,* 25(7), 702-713.

Studer, R., Benjamins, V. R. and Fensel, D. 1998. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering,* 25(1-2), 161-197.

Suh, E., De Weck, O. and Chang, D. 2007. Flexible product platforms: framework and case study. *Research in Engineering Design,* 18(2), 67-89.

Suh, M. S., Jhee, W. C., Ko, Y. K. and Lee, A. 1998. A case-based expert system approach for quality design. *Expert Systems with Applications,* 15(2), 181-190.

Suh, N. P., 2001. *Axiomatic design: advances and applications.* New York: Oxford University Press.

Takai, S. 2009. A case-based reasoning approach toward developing a belief about the cost of concept. *Research in Engineering Design,* 20(4), 255-264.

Tseng, Y.-J., Kao, Y.-W. and Huang, F.-Y. 2008a. A model for evaluating a design change and the distributed manufacturing operations in a collaborative manufacturing environment. *Computers in Industry,* 59(8), 798-807.

Tseng, Y.-J., Li, W.-P. and Lin, Y.-H. 2008b. Evaluation of Design Change Alternative Cases Using an AHP Approach to Evaluate Component Relationships of a Product. *Journal of the Chinese Institute of Industrial Engineers,* 25(5), 358 - 367.

Ullman, D. G., 2010. The mechanical design process, fourth edition. *McGraw-Hill series in mechanical engineering.* 4th ed. New York: McGraw-Hill.

Ulrich, K. T. and Eppinger, S. D., 2004. *Product design and development.* 3rd ed. Boston, Mass. ; London: Irwin McGraw-Hill.

United Nations Conference on Trade and Development, 2010. *UNCTAD handbook of statistics 2010.* New York : United Nations.

Watts, F. 1984. Engineering changes: a case study. *Production and Inventory Management Journal,* 25(4), 55-62.

Whitney, D. E. 1988. Manufacturing by Design. *Harvard Business Review,* 66(4), 83-91.

Williams, B. J. and Carver, J. C. 2010. Characterizing software architecture changes: A systematic review. *Information and Software Technology,* 52(1), 31-51.

Wright, I. C. 1997. A review of research into engineering change management: implications for product design. *Design Studies,* 18(1), 33-42.

Wu, M.-C., Lo, Y.-F. and Hsu, S.-H. 2008. A fuzzy CBR technique for generating product ideas. *Expert Systems with Applications,* 34(1), 530-540.

Xue, D., Cheing, S. Y. and Gu, P. 2006. Configuration design considering the impact of design changes on downstream processes based upon the axiomatic design approach. *Journal of Engineering Design,* 17(6), 487 - 508.

Xue, D., Cheing, S. Y. and Gu, P. 2008. Parameter design considering the impact of design changes on downstream processes based upon the Taguchi method. *Journal of Engineering Design,* 19(4), 299 - 319.

Zou, Y. and Lee, S.-H. 2008. The impacts of change management practices on project change cost performance. *Construction Management and Economics*, 26(4), 387 - 393.

APPENDIX I: SUMMARIES OF QUESTIONNAIRES FOR SYSTEM VALIDATION

System Modelling
Q1: Can system modelling method adopted in this system meet the current requirements for representation of product design? Answers and Comments: Interviewee 1: Yes, almost. This method models the physical interactional relationships between components very well and intuitively. I think there is one point missing in the functional requirement model is that the design constraints within the product design should be more clarified since it's an important part to be considered in product design. Interviewee 2: Yes, the SysML based modelling method can represent a product design in terms of functional requirement modelling and physical interaction modelling. However, the CAD modelling for physical structure is not adequately formalised in this tool. Interviewee 3: Yes, it provides a good coverage of functionalities for conceptual product design. Q2: Does the modelling method for interactions between the components capture sufficient information and help you understand the product design? Interviewee 1: Yes, the modelling method for interactional relationships between

components works very well. It clearly presents the inputs and output of each component and the connections.

Interviewee 2: Yes, it does help engineers understand the design of the product in a visual way. The relationships between them are clear.

Interviewee 3: Yes, it provides comprehensive information and helps me understand a product design.

Q3: Does the functional modelling method present enough information of the functional requirements and help you understand the product design?

Interviewee 1: Not exactly. Although the model clearly maintains the hierarchical relationships between functional requirements, it may be better if it can have more information regarding the interdependent relationships between them.

Interviewee 2: Yes, it provides enough information for each functional requirement and establishes the relationships with the components. It helps me understand the product design.

Interviewee 3: Yes, it helps me understand the design but the graphic symbols are not really very understandable for me.

Q4: Does the tool used for system modelling have a user friendly interface and easy to learn and use?

Interviewee 1: Yes, this tool is very similar the UML modelling tool and made easy to build the models.

Interviewee 2: Yes, after being well explained, I think the tool is easy to use. But as

I don’t have very much software experience, it may take some time for me to pick it up.

Interviewee 3: No really. I feel difficult to use the tool. Maybe some training is needed for users.

Q5: Do you feel that the tool works efficiently in modelling the functional requirements and component interactional relationships?

Interviewee 1: Yes, it is efficient.

Interviewee 2: Yes, it is efficient as long as the designer understands the product design well and familiar with the software tool.

Interviewee 3: Probably.

Change propagation analysis
Q1: Does the change propagation analysis process developed in the system provide sufficient functionalities for the engineering design change analysis process in your company? Interviewee 1: Yes, the functional relationships and the interactional relationships are provided from the models which provide a lot of information for change propagation analysis. However, I think the design constraints also need to be considered. Sometimes although the functional requirements are satisfied, the design constraints are also needed to be checked and any violation of design constraints may cause the design fail. Interviewee 2: Yes, I think the provided functionalities are enough for change propagation analysis Interviewee 3: Yes, in terms of the conceptual design, it does. But the manufacturing factors should also be considered. Q2: Does the composite matrix developed in the system carry adequate information for change propagation analysis and help you understand the problem? Interviewee 1: Yes, the composite matrix provides sufficient information for design change analysis. The method of presentation makes the analytical process easy to understand and follow. Interviewee 2: Yes, I think it carries adequate information for change propagation analysis. It can help me understand the change and its connections with others. However, the guideline provided for the change analysis process covers directly connected components and their functions. How about the indirectly connected ones. For example, a change to a component may change a so-called 'flow' in this method. The 'flow' may not affect the functions of that component but it may

change the output flow of the component. This output flow may further change other components which are not directly connected by the original flow. I know we can always find the routes to track it in the matrix but a clear guideline would be better.

Interviewee 3: Yes, in terms of the design stage I think the matrix provides sufficient information to understand the connections between components and their functions

Q3: Do you think the change propagation analysis process is effectively carried out by using the composite matrix?

Interviewee 1: Although the matrix is able to provide sufficient information regarding the relationships between components and functions, I am concerned that if a product is very complex the efficiency of the current method may be low.

Interviewee 2: The matrix method for change propagation analysis is well presented in the system. But it seems it will be very big and hard to use if there are a lot of functions and components. Is there any way to break down the matrix in that case?

Interviewee 3: I think a clearer instruction is needed to explain each step of the change propagation analysis since the connections make the process a little complex.

Q4: Do you think if it’s useful to use this change propagation analysis approach to finding out the design conflicts underlying?

Interviewee 1: Yes, I think this method is very useful. It can help clarify the interaction and connections between components.

Interviewee 2: I think it's useful and helpful to understand the design. It will be better if the matrix can be improved.

Interviewee 3: If a clearer instruction for processing the change propagation can be provided, I think this method would be useful. At least the information it presents is very necessary for engineering change analysis.

Q5: Is the tool developed for the composite matrix easy to learn and use?

Interviewee 1: Yes, it is.

Interviewee 2: Technically, it is not hard to use this tool.

Interviewee 3: The system is easy to use and understand. It just needs a clearer guideline for the process.

Design knowledge management
Q1: Do you think whether the knowledge repository developed in this system provides a sufficient functional coverage as an infrastructure for knowledge management? Interviewee 1: Yes, it provides a good functional coverage of knowledge management and knowledge reuse for engineering design change analysis. One thing I would like to suggest for the knowledge management system is that it would be better to integrate with other enterprise system, for example PDM. That is because nowadays most design information is stored in those similar systems in companies. It will save a lot of time if it’s possible to import design information directly from existing systems Interviewee 2: As far as I know, the knowledge module provides a good coverage of functionalities for engineering change management. It enables engineers to find and reuse previous design cases to solve current problems. The idea of finding design cases according to their semantic meanings is very innovative. I have noted that the ontology building function is supposed to be carried out by the knowledge manager. I would suggest letting engineers use this function as well. Therefore the coverage of the ontology definition will be better. Interviewee 3: This module seems to provide a good coverage of functions for storing and retrieving design knowledge from previous cases. In my opinion, I think the functional coverage is sufficient. Q2: Do you think the method for knowledge management in system is able to effectively capture design knowledge from engineers’ everyday work? Interviewee 1: Yes, I think this method is able to capture the design knowledge from

engineers' daily work. It would be more effective if the integration is implemented.

Interviewee 2: Yes, the knowledge management system works to capture design knowledge. With contributions from engineers over time, more and more design knowledge will be stored which will make the system even more powerful.

Interviewee 3: Yes, it seems like the system can manage to get design information and store them structured if the engineers are encouraged to contribute.

Q3: Do you think the way of retrieving design knowledge and design cases from the knowledge repository is useful and effective?

Interviewee 1: Yes, the approach to retrieving formalised knowledge is novel and automated. It can be very useful and helpful for engineers and can save them a lot of time if the semantic similarity algorithm is accurate enough.

Interviewee 2: Yes, this method can be very useful and effective for knowledge retrieval since the process is much automated.

Interviewee 3: It may take some time to successfully get this system work since there are a lot of design cases need to be formalised and stored. But the idea is very good and I think it will be useful and effective when the knowledge repository is well built.

Q4: Do you think overall the knowledge based design conflict solving method is helpful to engineers in dealing with engineering design changes?

Interviewee 1: Yes, I think overall it is a very good idea for finding reference solutions from previous design cases.

Interviewee 2: Yes, I think it is helpful.

Interviewee 3: Yes, I think so. But there is a lot of work to do in implementation.

Q5: Do you think the tool is designed well and easy to learn and use?

Interviewee 1: Yes, I think is easy to use this tool although some training is needed for the formalisation approach.

Interviewee 2: It is easy with tool developed in the system, very straightforward.

Interviewee 3: I don’t think this method is easy for me since the ontology management and rating is not really an easy job for me. Also the formalisation process may need more automation mechanisms for ordinary engineers.

Change solution evaluation
Q2: Do you think if this method provides sufficient functional coverage for evaluating the impact of an engineering design change? Interviewee 1: Yes, the functions of the method seem sufficient. The considerations of development time, development cost and development risk are useful from the viewpoint of project management. Interviewee 2: The way of evaluating change impacts is good. But the criteria selected in this method may not be sufficient in some projects. It would be a better idea if the criteria can be configurable by users. Interviewee 3: Yes, the functional coverage is relatively sufficient. Q2: Do you think information which is designed to be obtained from other enterprise systems is adequate for assessing the change solution? Interviewee 1: The web service approach to integrating with other enterprise systems great. Information obtained from other systems is adequate from my point of view. Interviewee 2: As I mentioned, information for evaluation may vary from project to project. A configurable criteria selection measure may work better. Interviewee 3: It actually depends on the project. For some projects, more detailed information may need, for example, manufacturing capability of the company. Q3: Do you find the change solution evaluation method is easy to use and learn?

Interviewee 1: Yes, it is easy to use if the integration is well established.

Interviewee 2: Yes, it is.

Interviewee 3: It seems an easy tool but the relative important calculation may take some time for designers.

APPENDIX II: PUBLICATIONS

The rest of section of appendix II presents the publications derived from this research project.

A Model-driven and Knowledge-based Methodology for Engineering Design Change Management

Genyuan Fei[1], James Gao[1], Dele Owodunni[1] and Xiaoqing Tang[2]

[1]University of Greenwich, J.Gao@gre.ac.uk
[2]Beihang University, xqtang@buaa.edu.cn

ABSTRACT

Change management is very important to the success of new product development. The earlier that change issues are addressed, the greater that product lifecycle costs can be saved. This paper presents a novel methodology that has been developed to help designers trace, analyse and evaluate engineering changes occurring in the product design phase. A modelling method is employed to enhance the traceability of potential design changes occurring between the functional and structural domains of design. Based on functional and physical models, a matrix is developed to analyse change propagations and help identify design conflicts arising from design changes. A knowledge based methodology has been developed to resolve design conflicts by reusing previous design change knowledge. A wind turbine for power generation from the collaborating company is used to evaluate the developed methodologies.

Keywords: engineering change, design conflict resolving, knowledge management.
DOI: 10.3722/cadaps.2011.373-382

1 INTRODUCTION

Engineering changes have been recognised as inevitable in complex engineering product development [1, 2]. They have great influence on downstream developing and production activities, making product development very costly and time consuming. Therefore, it is critical to keep them under control. On the other hand, engineering changes have also been recognised as a source of innovation and creativity which can facilitate evolutions of products and technologies [3-4]. From this perspective, knowledge acquired from engineering changes is also very useful to product development in the long term. Significant research has been reported in engineering change management (ECM) regarding computerising traditional paper-based engineering change processes [5], improving communication methods between engineers [6], clarifying knock-on change effects between components [7]. Most early research shows the efforts made in dealing with engineering changes in the manufacturing process. More recently, international research effort has been focusing on managing changes in the early stages of product lifecycle, as effects in the design stage may significantly cause increase of the total development cost. One of the most important issues in change management is change propagation which often leads to great complexity by spreading knock-on effects via dependency relationships between design elements. Some methods have been proposed trying to predict change propagations and analyse their potential impacts on other parts of the system [8]. However, engineering changes always propagate dynamically, which means that it is difficult to predict change propagations and how the dependency relationship within a system will change until the solution for the preceding change is determined. The complexity of engineering change analysis means that designers could be easily exhausted before having investigated the entire design space. Some IT systems have been developed to help designers by computerising the traditional ECM process [5], integrating with other manufacturing systems to enhance communications, and visualising change propagation tracking [9]. However, there is a lack of tools for resolving design conflicts arising from change propagations using previous design experience. It should be noted that engineering change management has been an important part of product lifecycle management (PLM). Researchers have developed and used PLM technologies to manage the information, knowledge and business processes related to a product in its whole lifecycle (e.g., a leading research group in features reported a PDM-based framework for collaborative aircraft design [10,11]. This project aims at dealing with changes in the functional domain and the physical structure domain of complex product design. It would help designers analyse design change propagations within these two domains. It would also help to resolve design conflicts arising from design changes using knowledge from previous design change cases. There are mainly three objectives, i.e., (i) dynamically capture changes and their propagations between functional requirements and physical components; (ii) identify and formalise design conflicts arising from design changes; and (iii) use knowledge based engineering (KBE) technology to facilitate finding solutions for design conflicts from previous design cases.

2 DESIGN CHANGE PROPAGATION ANALYSIS AND CONFLICT IDENTIFICATION

An important argument brought forward in this research is that *Change propagation is caused by design conflict that occurs when a change of a part of the system obstructs or harms realisations of the functions of other parts.* Design conflicts are quite common in product development, while designers work on respective parts of a system and do not have a complete picture of the dependencies between each part in early phases. However, even if the system has been successfully put together, design conflicts still happen when some parts of the system change. Although a lot of methods have been proposed to predict change propagations in engineering change management, there is a lack of methods for formalising the change propagation chain. When considering design conflicts arising during change propagations, the dependencies between design elements should be further analysed. The change propagation process can be broken down as shown in Fig. 1 so that the impact of each phase of the propagation chain can be analysed effectively.

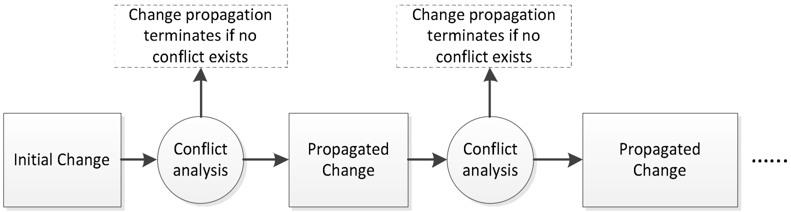


Fig. 1: Process of change propagation analysis.

When the initial change is determined, designers need to analyse whether the result of this change may obstruct or harm realisations of functions of other parts of the system. If it does, then further changes need to be carried out. These are the so-called propagated changes. If it causes no negative influences on other parts and they can function well as designed, then change propagation ends. The cooling system of a new wind turbine (Fig. 2) is used in this project to evaluate the proposed methodology. This product is being developed by the industrial collaborator which is a pioneer in gearless wind turbine development. Nearly 2000 wind turbines (by May 2010) based on their solutions have been deployed in Europe, Asia and America. An example change in design requirement is that a customer needs the wind turbine to be deployed in particular sandy environments, and the air filtering of the cooling system needs to be much improved to prevent heavy sands coming to the cooling system. This change causes some knock-on effects that give rise to changes on other parts of the system. The methodology proposed in this paper aims to solve this problem by modelling the cooling system, identifying design changes and related design conflicts, and resolving the chosen design conflicts using a knowledge-based system.

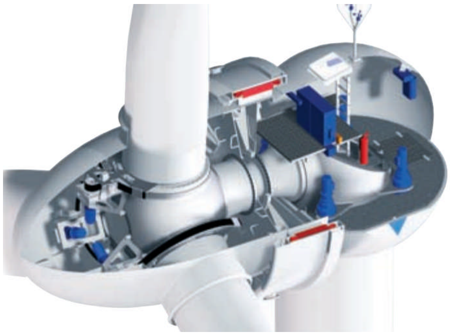
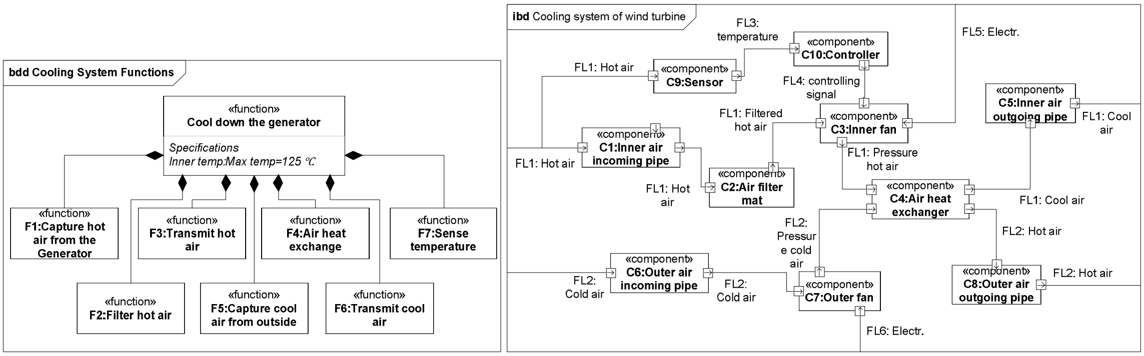


Fig. 2: An inside view of the wind turbine (Courtesy of Vensys AG Engergy).

2.1 Functional and Physical Dependency Modelling

Before going into design change propagation analysis, it's important to clarify dependent relationships between elements of engineering design. Elements of an engineering design considered in this project include functional requirements in the functional domain, and physical components in the physical domain. Consequently, dependent relationships include dependencies between functional requirements, involvement of the physical components for the realisations of functional requirements, and interactional and spatial relationships between physical structures. The block definition diagram (BDD) of SysML™ [12] is used to model the functional requirements (Fig. 3a), the internal block diagram (IBD) of SysML™ is used to model the interactional relationships between physical components (Fig. 3b), and the CAD geometric models are used to clarify spatial relationships between physical components. The composite matrix proposed in the following section summarises the interactional relationships depicted in Fig. 1 (b). It also clarifies the mapping relationships between functional requirements and physical components, and the spatial relationships between physical components.



a: Functional analysis of cooling system

b: Analysis of interactions in the cooling system

Fig. 3: Modelling methods for functional and physical dependencies.

2.2 The composite Matrix for Design Change Propagation Analysis

A matrix based method is employed to analyse change propagations within and between functional requirement domain and physical structure domain. A composite matrix (see Fig. 4) is constructed based on the results of modelling analyses of functional structure, physical interaction and physically spatial relationship. The matrix is composed of three parts which are marked by different colours. The first part (the green part) represents the mapping relationship between function and components. Elements in the first column represent physical components, and elements in the green part of the first row represent functions. Each marked cell in the green part represents the involvement of a component in the realisation of a function. The second part (the blue part) represents the interaction relationship between components, which reflects the modelling results of physical interactions between components. Elements in the first row of the blue part represent flows in Fig. 3(b). When there are changes happening on a component (an element in the first column), related flows going through it will be identified in the matrix. Components that these flows go through are also identified. The third part (the grey part) represents the physically spatial relationship between components. A marked cell in this part the matrix means the component in the column is physically connected with the component in the row. Clarification of this type of relationship helps designers find potential propagating changes to neighbouring components. Although in some research, matrix-based methods have been discussed for static dependency analyses between physical components in product design, this proposed composite matrix can be useful to dynamically analyse dependencies between the functional domain and physical domain of product design and facilitate change propagation capture. The process of how to use the composite matrix has been discussed in section 2.3.

			Funtions							Flows						Components									
			F1: Capture hot air from the Generator	F2: Filter hot air	F3:Transmit hot air	F4: Air heat exchange	F5 :Capture cool air from outside	F6: Transmit cool air	F7: Sense temperature	FL1: Air flow from the wind turbine to be	FL2: Air flow from outside for cooling	FL3: Temperature signal flow	FL4: Controlling signal flow	FL5: Electrical energy flow for the inner fan	FL6: Electrical energy flow for the outer fan	C1:Inner air incoming pipe	C2:Air filter mat	C3:Inner fan	C4:Air heat exchanger	C5:Inner air outgoing pipe	C6:Outer air incoming pipe	C7:Outer fan	C8:Outer air outgoing pipe	C9: Heat Sensor	C10:Controller
			wf(1)	*wf(2)*	*wf(3)*	*wf(4)*	*wf(5)*	*wf(6)*	*wf(7)*																
Components	C1:Inner air incoming pipe	*wc(1)*	X		X					X							X	X	X					X	
	C2:Air filter mat	*wc(2)*		X						X						X									
	C3:Inner fan	*wc(3)*	X		X					X			X	X		X				X					
	C4:Air heat exchanger	*wc(4)*				X				X	X						X				X				
	C5:Inner air outgoing pipe	*wc(5)*			X					X															
	C6:Outer air incoming pipe	*wc(6)*					X	X			X														
	C7:Outer fan	*wc(7)*					X	X			X				X					X	X				
	C8:Outer air outgoing pipe	*wc(8)*						X			X							X							
	C9: Heat Sensor	*wc(9)*	X						X			X													
	C10:Controller	*wc(10)*	X									X	X												

Fig. 4: Composite matrix for change analysis.

2.3 Process of Design Change Propagation Analysis and Design Conflict Identification

This section clarifies the idea of analysing change propagations and identifying design conflict arising from change propagations. The method is described in association with a special customer requirement (i.e., a change) of improving the air filtering measure mentioned above and is based on the composite matrix for change analysis (Fig. 4). The change is triggered by a functional requirement called 'F2: Filter hot air'. The analytical process starts from the function-component part of the matrix.

Step 1: Identify component changes caused by functional change. As mentioned above, because of the sandy environment where the wind turbine will be deployed, the current air filtering measure cannot meet the requirement. Fig. 4 shows components involved in the realisation of function, filter hot air (F2). Only one component C2 is identified. To meet the sandy environment, the current air filter mat with a dust holding capacity 650g/m^2 needs to be increased to 750g/m^2.

Step 2: Identify potential affected components. The component changed in the above step may change the physical statuses of flows going through it and may also change its neighbouring components due to changes of its spatial characteristics. Led by component C2, the row shows flows and neighbouring components that are potentially affected by the change of C2. In this case, flow FL1 and neighbouring component C1 are related to C1. The flow FL1 also goes through C1, C3, C4, C5, so these 4 components may also be potentially affected by the change of C1. The side effect of changing the air filter mat is that the mat with higher dust holding capacity is thicker and causes larger air pressure drop, which can significantly reduce the efficiency of heat exchanging.

Step 3: Check change effects with related functions. Components that are affected by the flows and the spatial connections need to be checked whether the changed flows or the changed spatial connections would affect the realisations of their related functions. In this case, the air flow after the filter mat has a lower pressure which means components C3, C4 and C5 would be potentially affected since the status of air through them is changed (see the column led by FL1). According to the analysis by engineers, the lower air pressure through C3 will weaken its performance. Also the lower air pressure through C4 will reduce the efficiency of heat exchange. But it has almost no effect on C5. The spatial change (thicker filter mat) has been considered as not noticeable change on C1 since the change can be easily accommodated by the current design. Although in this case change caused by spatial connection is negligible, in many other cases it may be significant and corresponding changes need to be made. For example, in this case, change the component C1 or add some other structures to accommodate the changes caused by spatial connections. Therefore, C3 and C4 have been identified as affected components which need to be changed to accommodate negative impact from the previous change on C2.

Step 4: Identify and solve design conflicts. By analysing affected components, design conflicts can be identified. Taking C4 as an example, the changed input flow is the incoming air (FL1). Its pressure is lowered due to the change of the air filter mat (C2) and the affected parameter is the heat exchange efficiency which is also lowered. This effect means the heat exchange cannot meet the functional requirement F4. Therefore this design conflict needs to be solved. A knowledge based method is developed to help designers find reference solutions from previous design cases. The ways of formalising design conflict and reasoning in the knowledge base have been presented in the next section.

Step 5: Analyse change propagations caused by component changes in step 4. When a proper solution has been found in step 4, changes on affected components have been determined. These changes would potentially affect other components as well in form of design conflict. In the above case, if component C4 has been changed flows FL1, FL2 and connected components C2, C6 may also be potentially affected. Thus, a next round of change analysis also needs to be carried out until there is no further change conflict being identified, which means change propagation stops and change analysis initiated by the first change can be finished.

3 KNOWLEDGE-BASED DESIGN CONFLICT RESOLVING

The contradiction matrix and invention principles of TRIZ have been used by many companies to resolve technical conflicts [13]. The idea of TRIZ to solve conflicts includes generally four steps: (1) identify technical conflicts; (2) generalise technical conflicts by using 39 engineering parameters; (3) find invention principles via a standard contradiction matrix; and (4) explore specific solutions by following the indications of invention principles. Although problem solving techniques of TRIZ are innovative and inspirational to engineers, the method is difficult to master without comprehensive training and long-time experience. The authors proposed a knowledge-based method which works in a similar way but more intuitive and easier to use. Fig. 5 depicts the process of how design conflicts have been solved.

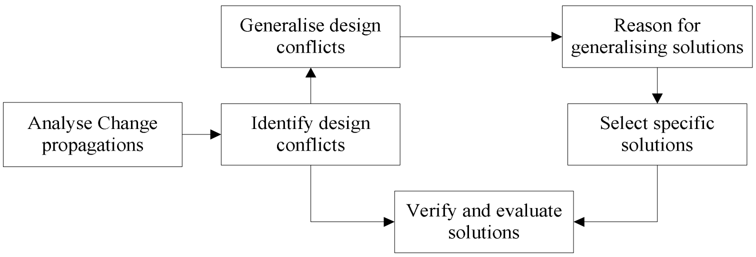


Fig. 5: Process of resolving design conflicts.

Design conflicts, identified through the matrix-based design change propagation analysis, need to be formalised and generalised with predefined ontology which is also the basis of the knowledge system. Fig. 6 shows the formalisation of a design conflict caused by flows. A flow-caused design conflict includes physical components where the conflict happens, the changed input flows that cause the conflict and affected output flows that are affected by the changed input flows. Both the input flows and the output flows are formalised by the flow ontology and the characteristics ontology. The flow ontology defines the type of the flow. The characteristics ontology defines the key properties of a flow. For example, the gas flow normally has properties like pressure, temperature and moisture. Properties formalised in this part should be critical to the operation of the component. The behaviour ontology defines how the flow is changed, which is used with the characteristics and/or other flows or objects.

The component is formalised by the functional ontology. The functional ontology defines what the component does and what object the component uses while it operates. The characteristics are critical for the performance of the operation of the component. For example, in the cooling system, there are two characteristics of the heat exchanger are important to its functionality. One is the area of the heat exchanger surface. Wider surface can have higher heat exchanging efficiency. The other one is the material that the heat exchanger is made of. Some material, for example bronze, has a better heat conduction performance than others (e.g., steel).

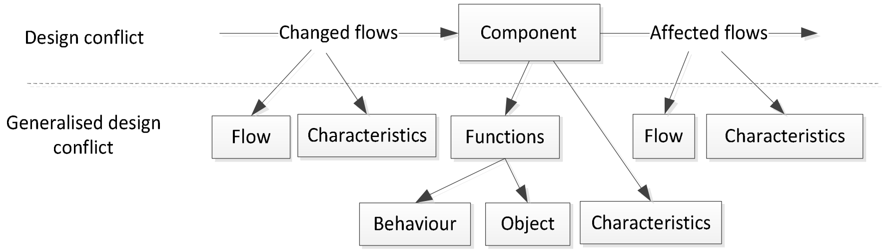


Fig. 6: Formalisation of design conflicts.

The functional ontology used, so called functional basis, contains generalised functions and flows which are regarded as a useful and comprehensive engineering functional ontology. The functional ontology includes *behaviour* ontology and *flow* ontology. *Domain* ontology is also developed to classify products, subsystems and components used in the wind turbine. Protégé is used as an ontology editor to develop the proposed ontology. Protégé was developed by Stanford University and is the de facto in the academic community for ontology development. Fig. 7 shows some parts of the ontology, including definitions of flows, definitions of functions, definitions of product components and an instance of a flow in the cooling system.

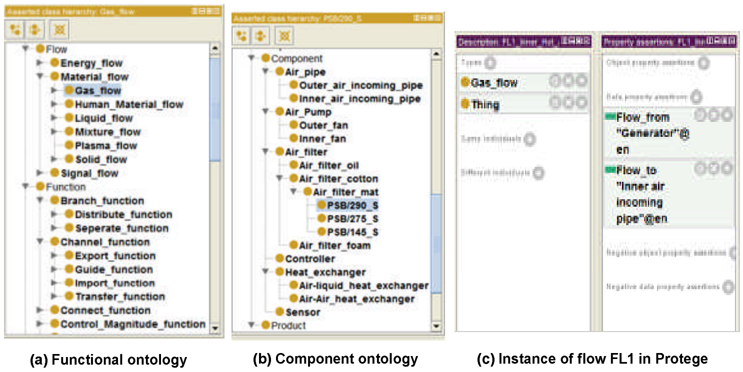


Fig. 7: Ontology developed for engineering design change.

When a design conflict arising from design change analysis, it is formalised and generalised by using defined ontology to specify the essences of input flows, output flows, and functionalities. The generalised model of the design conflict will then be pushed into the knowledge system. The system will reason in the knowledge repository by analysing the semantic similarities between different concepts to find most similar functional and physical descriptions. After that, solutions attached to these descriptions will be retrieved as reference solutions for the current design conflict. Designers can adjust or adopt similar solutions to solve current problems.

The framework of knowledge based system for design conflict resolving is shown in Fig. 8. In this system, design cases are collected from many sources including different functional departments and IT systems. The design cases are formalised in a hierarchical way, which clarifies functions or problems a design case is to address, the solutions used in this design case, the components involved in this solution and characteristics that contribute to the realisation of the function or the problem solving. The formalised structure is also been generalised by domain ontology including behaviour ontology, flow ontology, subsystem/component ontology and physical characteristic ontology. Therefore, a design conflict generalised by the same set of ontology is able to find reference solutions by comparing semantic similarities and retrieve related and useful previous design cases.

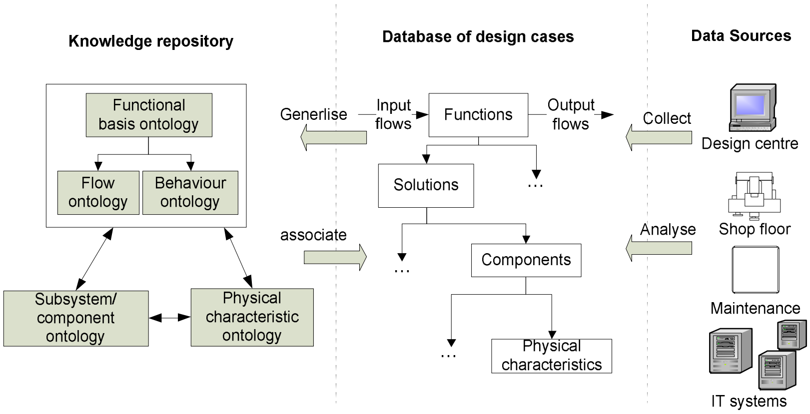


Fig. 8: Knowledge repository building using previous design cases.

A three-tier web-based prototype system is developed implement knowledge-based system for engineering design change management. Software systems involved in this prototype include MySQL as an infrastructure of data storage, Tomcat 6.0 as a web server and servlet container, Java enterprise edition (Java EE) as the business implementation architecture, and also Java Server Page (JSP) as a presentation technology. Protégé is deeply used to edit ontology, convert and store ontology into a relational database. Integrations with other business systems like PDM, ERP and SCM are going to be done at the next stage of this research.

4. REASONING METHOD FOR DESIGN CONFLICT SOLVING

The reasoning method is critical to finding candidate solutions for design conflicts arising from design change propagation analysis. It works with generalised design conflicts and the knowledge repository. Both are formalised based on the predefined set of ontology. The reasoning method is basically composed of two steps: (i) The generalised design conflict is used to compare with formalised design cases stored in the knowledge repository and find semantically similarly general solutions; and (ii) experienced engineers review retrieved design cases associated with general solutions to find the most viable ones. The general approach to solving design conflicts is depicted in Fig. 9.

In the proposed reasoning methodology, one of the most important steps is to analyse semantic similarities between generalised design conflicts and general solutions stored in the knowledge repository. As described above, design conflicts are generalised by predefined ontology and also previous design cases are formalised using the same set of predefined ontology. If a design conflict is defined as a definition (D) and all of the stored general solutions are defined as a set $\{D_1 \dots D_n\}$, the first step is to find the most similar solutions from the set of general solutions. Both the design conflicts and the general solutions are formalised by the same set of ontology. Each element of the design conflict and the design case is tagged by an ontological concept of the predefined set of ontology (a node in the hierarchical structure). The algorithm of calculating the semantic similarities between a generalised design conflict and general solutions is comparing the semantic similarity of each corresponding element (e.g. the flow type of the changed incoming flow in Fig. 6) and then adding them up to get an overall semantic similarity.

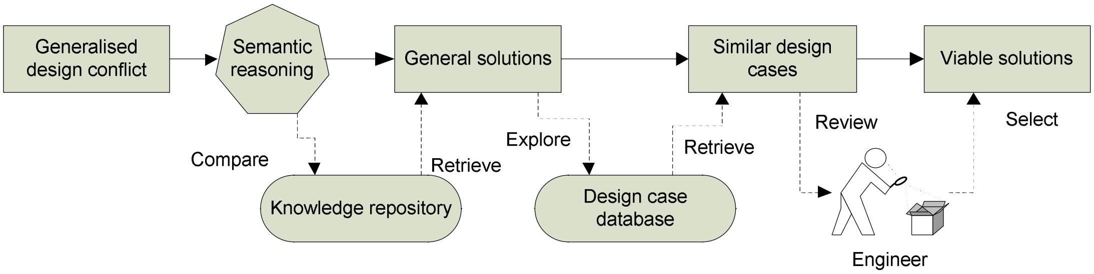


Fig. 9: Reasoning approach to design conflict solving.

Fig. 10(a) represents hierarchically ontological definition of a group of concepts. The higher of levels a concept stays in, the more general semantic meaning it represents. While in lower levels, the semantic meaning of a concept is more specific. In Fig. 10(b), IC(S1) represents the semantic meaning of the concept S1. Since S1 is the parent of S11 and S12, S1 has a wider semantic meaning than S11 and S12, which means:

$$IC(S11) \in IC(S1), and\ IC(S12) \in IC(S1) \qquad (1)$$

Theoretically, the following equation can represent how S11 (a child) is semantically similar to S1 (a parent) by comparing scales of semantic meaning of each concept:

$$Sim(S11, S1) = IC(S11)/IC(S1) \qquad (2)$$

While the similarity of S11 (a brother) to S12 (a brother) can be represented as:

$$Sim(S11, S12) = (IC(S11) \cap IC(S12))/IC(S12) \qquad (3)$$

Thus, the similarity of any two concepts (for example, S111 and S22) can be represented as:

$$Sim(S111, S22) = \{Sim(S111, S11) \times Sim(S11, S1) \times Sim(S1, S2) \times Sim(S22, S2)\} \qquad (4)$$

While in reality, it is well-known that exact semantic meaning of a concept is very difficult to measure. More often, people can tell the qualitative similarity between two concepts by their experience and common sense. So in this research, a survey is developed and experienced engineers are asked to rate semantic similarities between child-concepts and their direct parent-concepts (e.g. $Sim(S11, S1)$), and also between their brother-concepts (e.g. $Sim(S11, S12)$).

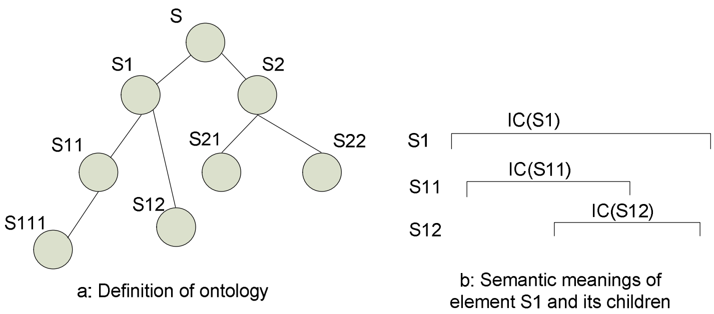


Fig. 10: Ontology definition and information content.

Based on the ideas of rating and calculating similarities between concepts, the generalised deign conflict can be compared with general solutions in the knowledge repository, since both of them are formalised using the same set of ontology. The formalisation of general problems that general solutions intended to solve is the same as for design conflict formalisation. So the similarity between a generalised design conflict and a general problem can be described as:

$$Sim(DC, GP) = \{Sim(CF_{DC}, CF_{GP}) \times Sim(Function_{DC,} Function_{GP}) \times Sim(AF_{DC}, AF_{GP})\} \quad (5)$$

Where DC represents design conflict, CF represents general problem, CF represents changed flow, and AF represents affected flow. Similarity between changed flows can be represented as:

$$Sim(CF_{DC}, CF_{GP}) = Sim(Flow_{DC}, Flow_{GP}) \times Sim(Character_{DC}, Character_{GP}) \quad (6)$$

Similarity between functions can be represented as:

$$Sim\left(Function_{DC,} Function_{GP}\right) = Sim(Behaviour_{DC}, Behaviour_{GP}) \times Sim(Object_{DC}, Object_{GP}) \times Sim(Character_{DC}, Character_{GP}) \quad (7)$$

Similarity between affected flows can be represented as:

$$Sim(AF_{DC}, AF_{GP}) = Sim(Flow_{DC}, Flow_{GP}) \times Sim(Character_{DC}, Character_{GP}) \quad (8)$$

By comparing the overall similarities between the generalised deign conflict and general problems, a set of prioritised similarity values are generated:

$$\{Sim(DC, DP_1), Sim(DC, DP_2), \ldots, Sim(DC, DP_n)\}$$

By exploring and reviewing design cases associated with general problems (corresponding to general solutions) from higher priority to lower priority, the suitable design cases are chosen as reference solutions for the target design conflict. An example of resolving the design conflict identified in the section of change propagation analysis has been developed using the proposed reasoning method. The result is presented in a tabular form in Fig. 11.

Design conflict description			Generalised conflicts				General solutions	Design cases
Change	Cause	Effect	Formalisation		Generalisation by Ontology			Design documents
Change Air Filter Mat	Lower air pressure (FL1) from the inner air incoming pipe (C1)	Reduce the efficiency of heat exchanging (C4)	Affected Function	Air heat exchange (F4)	Behaviour:	Conduct	1, Change the material of the conduction part; 2, Increase the area of conduction interface; 3, Improve the structure of air vessel; 4, Increase the volecity of air flow	1, Passive cooling system of the unclear reactor (File no: ****) 2, Patent: DE 19636591C; 3, Central air conditioner (File no. *****);
					Object:	Thermal flow, Gas flow		
			Changed input flow	Air flow from incoming pipe(FL1) with lowered pressure	Flow:	Thermal flow, Gas flow		
					Characteristics	Pressure		
					Change:	Reduce		
			Affected output flow	External hot air from the exchanger (FL2) with less heat exchanged	Flow:	Thermal flow, Gas flow		
					Characteristics	Temperature		
					Change:	Increace		
			Affected Component	Air Heat Exchange (C4), Nacelle	Air-Air heat exchanger			
			System	Wind turbine cooling device	Cooling system			

Fig. 11: Example for design conflict resolving.

5 CONCLUSIONS

Any design changes either in functional requirement domain or in the physical structure domain will potentially affect operations of other parts. Change propagations and their impacts are difficult to capture, which makes product design in uncertainty. The authors argue that design change propagation is caused by deign conflicts arising from the initial and consequent changes which are difficult to predict without knowing their preceding change solutions, since all the following changes are based on the solutions of preceding changes. Knowledge from previous design change cases is an important asset for companies. Many design conflicts arising from change analysis can be tackled by reusing well-formalised and managed knowledge abstracted from previous design cases. In this paper, a methodology for design change management has been proposed to implement these ideas using modelling method, matrix analytical method and knowledge based engineering. First, the system engineering modelling method captures dependency relationships in and between functions and components. The matrix-based analytical method helps to trace change propagations and identify design conflicts by following mapping relationships between functions and components, and behavioural and spatial connections between components. Using the design conflicts formalisation approach, design conflicts can be formalised based on functional ontology, system/component ontology and physical characteristic ontology, which makes knowledge reasoning in the knowledge repository possible. The framework of the knowledge repository is proposed to collect knowledge of design change from previous design cases. With help of the knowledge repository and the reasoning method, designers are able to find proper solutions for design conflicts occurring in design change propagation.

REFERENCES

[1] Huang, G.Q.; Yee, W.Y.; Mak, K.L.: Current practice of engineering change management in Hong Kong manufacturing industries, Journal of Materials Processing Technology, 139(1-3), 2003, 481-487. doi:10.1016/S0924-0136(03)00524-7

[2] Keller, R.; Eckert, C.M.; Clarkson, P.J.: Using an engineering change methodology to support conceptual design, Journal of Engineering Design, 20(6), 2009, 571 - 587. doi:10.1080/09544820802086988

[3] Balogun, J.; Jenkins, M.: Re-conceiving Change Management: A Knowledge-based Perspective, European Management Journal, 21(2), 2003, 247-257. doi:10.1016/S0263-2373(03)00019-7

[4] Eckert, C.; Clarkson, P.J.; Zanker, W.: Change and customisation in complex engineering domains, Research in Engineering Design, 15(1), 2004, 1-21. doi:10.1007/s00163-003-0031-7

[5] Huang, G.Q.; Yee, W.Y.; Mak, K.L.: Development of a web-based system for engineering change management, Robotics and Computer-Integrated Manufacturing, 17, 2001, 255-267. doi:10.1016/S0736-5845(00)00058-2

[6] Shiau, J.-Y.; Wee, H.M.: A distributed change control workflow for collaborative design network, Computers in Industry, 59(2-3), 2008, 119-127. doi:10.1016/j.compind.2007.06.010

[7] Ariyo, O.O.; Eckert, C.M.; Clarkson, P.J.: Challenges in identifying the knock-on effects of engineering change, International Journal of Design Engineering, 2, 2009, 414-431. doi:10.1504/IJDE.2009.030821

[8] Kocar, V.; Akgunduz, A.: ADVICE: A virtual environment for Engineering Change Management, Computers in Industry, 61(1), 2010, 15-28. doi:10.1016/j.compind.2009.05.008

[9] Ouertani, M.Z.; Gzara, L.: Tracking product specification dependencies in collaborative design for conflict management, Computer-Aided Design, 40(7), 2008, 828-837. doi:10.1016/j.cad.2007.07.002

[10] Li, Y.G, Ding, Y.F., Mou, W.P. and, Guo, H., Feature recognition technology for aircraft structural parts based on holistic attribute adjacency graph, Proceedings of the Institution of Mechanical Engineers, Part B, Journal of Engineering Manufacture, 2010, 224(2):271-278

[11] Li, Y.G., Pan, Z.Y., Yan, R.J. and Liao, W.H., A PDM-based framework for collaborative aircraft tooling design, International Journal of Production Research, 2008, 46(9): 2413-2431

[12] Object Management Group. *OMG Systems Modeling Language (OMG SysML) V1.1*. 2008, http://www.omg.org/spec/SysML/1.1/PDF.

[13] Altshuller, G.S.: And suddenly the inventor appeared : TRIZ, the theory of inventive problem solving, Worcester, Mass., 1996.

International Journal of Computer Integrated Manufacturing
Vol. 24, No. 6, June 2011, 535–551

A method for engineering design change analysis using system modelling and knowledge management techniques

Genyuan Fei[a], James Gao[a]*, Oladele Owodunni[a] and Xiaoqing Tang[b]

[a]*Centre for Innovative Product Development, School of Engineering, University of Greenwich, Chatham Maritime, Kent, UK;* [b]*School of Mechanical Engineering and Automation, Beihang University, Beijing, China*

(*Received 2 September 2010; final version received 19 January 2011*)

Engineering design change management is very important to the success of engineering product development. It has been recognised that the earlier change issues are addressed, the greater product lifecycle costs can be saved. However, in practice, most engineering changes happen in the manufacturing phase, the later phase of product development. Change issues happening in the design phase, especially between the functional and the structural domains, have been a research focus in recent years, and thus there is significant research work that has been carried out to resolve early engineering change issues from different perspectives. This article presents a novel methodology that has been developed to help designers trace, analyse and evaluate engineering changes occurring in the product design phase. A modelling method is employed to enhance the traceability of potential design changes occurred between the functional and structural domains of design. Based on functional and physical models, a matrix-based method is developed to analyse change propagations between components and help find out design conflicts arising from design changes. A knowledge-based method has been proposed to resolve design conflicts by reusing previous design change knowledge. An industrial example about changes of a wind turbine cooling system has been used to help understand the methodology and prove its usefulness.

Keywords: engineering design; change management; system modelling; design conflict solving; knowledge management system

1. Introduction

Engineering changes have been recognised as inevitable in complex engineering product development (Huang *et al*. 2003, Palani Rajan *et al*. 2005, Keller *et al*. 2009). They have great influences on downstream developing and production activities, thus making product development very costly and time consuming (Huang *et al*. 2003). Therefore, it is critical to keep them under control. Engineering changes have also been recognised as a source of innovation and creativity that can facilitate evolutions of products and technologies (Balogun and Jenkins 2003, Jarratt *et al*. 2003, Eckert *et al*. 2004). From this perspective, knowledge acquired from engineering changes is also very useful to product development in the long term. Despite the different perspectives, both of the two arguments reflect the importance of engineering change management (ECM) in product development.

In the past, a lot of research have been done in ECM regarding computerising traditional paper-based engineering change processes (Huang *et al*. 2001), improving communicating methods between engineers (Shiau and Wee 2008), clarifying knock-on change effects between components (Clarkson *et al*. 2004, Eckert *et al*. 2006). Most early research shows the efforts made in dealing with engineering changes in the manufacturing process. Recently, changes happening in the critical stage of product development, the product design stage, have been emphasised (Mckay *et al*. 2003). It is recognised that the design stage of product development could determine the largest cost savings during the product life cycle. This means, changes happening at the design stage would have a greater impact than those happening in the manufacturing phase. This project focuses on analysis of changes and their propagations in the product design phase and solving design conflicts arising from them by using knowledge management technologies.

As an important part of product development, ECM has been studied by many academia and industrial practitioners in the past decade. They have identified issues within ECM and tackled them with proposed solutions from different perspectives. An introduction of the concept of ECM and an overview of previous research are given below.

*Corresponding author. Email: j.gao@gre.ac.uk

ISSN 0951-192X print/ISSN 1362-3052 online

DOI: 10.1080/0951192X.2011.562544
http://www.informaworld.com

1.1. The concept of engineering change management

Although engineering change has been studied for many years, its definition varies according to statements of different researchers. Wright (1997) defined engineering change as modification to a component of a product before it goes into production. Some researchers agree that engineering change is modification to dimensions, fits, forms, functions and materials to product or components after the product design is released (Huang *et al*. 2003, Kocar and Akgunduz 2010). Whilst some other researchers view engineering changes as changes that occur in a wider range from customer requirements to product in use (Pikosz and Malmqvist 1998, Eckert *et al*. 2004).

While research focuses have been shifting overtime, the scope of research on engineering change has been widened. The initial motive of studying ECM was to avoid engineering changes during the manufacturing process due to the adverse effects they cause. The adverse effects caused in terms of delivery time, developing cost and product quality are noticeable but very difficult to estimate (Huang *et al*. 2003). Later on, people realised that engineering changes are actually inevitable. Therefore, researchers have turned to finding out how engineering changes go on and what kind of impacts they may cause (Clarkson *et al*. 2004, Ouertani 2008, Kocar and Akgunduz 2010). Recently, some researchers argue the benefit of engineering changes to innovation and creativity, which can enhance the competitiveness of companies. Thus, some researchers have started to study engineering changes from perspectives of knowledge management and knowledge reuse (Balogun and Jenkins 2003, Palani Rajan *et al*. 2005, Lee *et al*. 2006, Keese *et al*. 2009).

The process of organising engineering change activities has also been explored in the past decade, in order to find most efficient and effective approach of ECM. A general process of engineering change has been proposed by Clarkson and Eckert (2005). This process includes six steps, namely engineering change request, possible solution identification, risk/impact assessment, solution selection and approval, solution implementation and change process review. Although the general process of engineering describes a reasonable approach to addressing change issues in product development, in reality, different companies have quite different processes to deal with engineering changes in order to fit their specific organisational and production requirements (Pikosz and Malmqvist 1998, Huang *et al*. 2003, Eckert *et al*. 2004).

1.2. Change propagation analysis

One of the most difficult issues in analysing engineering change is that when a component is changed, it may also change its related components (Ariyo *et al*. 2009). Therefore, an initial change may cause changes spreading at several structural levels. Essentially, the reason why changes propagate is because of the dependency between components. This situation is so-called change propagation or the knock-on effects of changes, which makes change analysis very tricky. Some researchers have made some efforts in dealing with this issue.

Clarkson *et al*. have proposed a method called change prediction matrix (CMP) to trace change propagations and analyse the impacts they may cause (Clarkson *et al*. 2004). The method transforms the dependency relationship between components in a product model to a design matrix. Based on this matrix, the likelihoods that potential change propagations may happen between components are estimated. Also in the same way, the impacts these potential change propagations may cause have also been estimated. By combining the change, likelihood matrix and the change impact matrix, a change risk matrix has been generated. With the help of visualising method, change propagation paths and their relative risks have been clarified.

In another study, Eckert *et al*. (2004) have proposed a method to analyse change propagation at a parametric level and identify four types of change propagation behaviours, namely constants, absorbers, carriers and multipliers. These four types of change propagation behaviours help to analyse change propagations that cross multi-levels. Four types of change propagation behaviours represent four situations when a change of a component propagating to another component via some other components, which includes changes being passed without effect, being reduced or eliminated, being replaced with new changes from the intermediate component and being enhanced. This method has also been integrated with the CMP method to enhance the performance of change propagation analysis in product conceptual design (Keller *et al*. 2009).

Kocar and Akgunduz (2010) have proposed a different method to analyse change propagations. They use visualisation technique and data mining technique to represent product models and find out dependencies between components. Users would be warned visually if potential change propagation is predicted to happen.

Ouertani (2008) has also proposed a visualisation tool called DEPNET to model product data and their dependencies within them. By using the product data dependent relationships, changes emerging during collaborative design process would be tracked down. Do *et al*. (2008) have also proposed a method for tracking engineering change propagation between different product data views based on a shared base product model. The method reduces data redundancy in ECM and maintains consistency between different product data views.

1.3. Information systems for ECM

Although methods for ECM have been proposed with rigorously analytical or reasoning approaches, designers will be easily exhausted before having investigated the entire search space (Ariyo *et al.* 2009). Therefore, it is important that computer-aided tools have been developed and used in ECM. Previous investigations have shown that computer-aided ECM systems have been rarely utilised in companies (Pikosz and Malmqvist 1998, Huang *et al.* 2003). Although some companies have used electronic document management systems to replace paper-based ECM documents, the data are non-structural and it is difficult to semantically trace similar engineering change cases.

Although currently not many companies are using computer-aided systems to facilitate their ECM, there are some systems that have been developed by academia trying to enhance communication and information sharing in change management process. Huang *et al.* (2001) developed a web-based system to implement the whole process of ECM, including engineering change log, engineering change request, engineering change evaluation and engineering change notice. The distributed system has improved the efficiency of ECM and enhanced the collaborations between engineers. Also, structural ECM data make it possible to integrate with other computer-aided systems such as product data management (PDM), enterprise resource planning (ERP), computer-aided design (CAD) and supply chain management (SCM). Ouertani and Gzara (2008) have developed a system called DEPNET to visually track dependencies within product specifications, so that change propagations can be captured if any design changes of a product specification happen. As mentioned above, Kocar and Akgunduz (2010) developed a visualisation system to track change propagation, which is well integrated with 3D modelling system. Lee *et al.* (2006) developed a knowledge-based system to facilitate ECM in a collaborative environment. The authors have used ontology technology and case-based reasoning method to construct a knowledge base of previous development experience. It also implements the knowledge base with a web-based system that enables users go through the whole ECM process from change request initiated to change approved.

1.4. Aim and objectives of the project

Based on previous research reviewed by the authors, some gaps in ECM have been identified. First, changes of functional requirements should be considered together with changes in the physical domain in the design phase. For example, in the domain definitions of the product design stage in the theory of Axiomatic Design (Suh 2001), when the changes in physical components are considered, the changes in the functional domain and their effects on the physical domain have not been considered. Second, there is a lack of consideration of the impact of change solutions on the change propagation analysis. A lot of efforts have been made to predict change propagations with predefined component interactions. However, specific solutions for change requests may dramatically change predefined interacting relationships between components, which may make predictions of later changes fail. Third, there is a lack of tools to help engineering designers reuse knowledge from previous cases regarding design change management in industry. In many design change cases, technical solutions for a design change request, or say for some similar design change requests, may have been re-developed on many other occasions. That may be because experience or technical solutions from previous design change cases have not been formalised and shared effectively.

This project therefore aims at dealing with changes in the functional domain and the physical structure domain of complex product design. It would help designers analyse design change propagations within these two domains. It would also help to solve design conflicts arising from design changes by reusing knowledge from previous design change cases. There are mainly three objectives, i.e. (1) dynamically capture changes and their propagations between functional requirements and physical structures; (2) identify and formalise design conflicts arising from design changes and (3) use knowledge-based engineering technology to facilitate finding solutions for design conflicts from previous design cases. A method for design change management is proposed by putting the emphasis on analysis of changes in the functional requirement domain and the physical structure domain. A modelling method is employed to enhance the traceability of changes occurring between functional model and structural model. A matrix-based method is constructed to capture dynamic change propagations between the two domains. In the end, a knowledge-based method is developed to help to solve conflicts arising during design change analysis. An industrial example has been used to show how the method works.

2. Analysis of design change management (DCM) practices

Referring to the literature reviewed above, most researchers focus on the engineering change analyses among physical components. Although some of them mentioned functional requirements would be influenced by changes of physical components, detailed discussion has not been made regarding how these influences happen and how to deal with them. In this section, changes taking place between the functional domain and the physical structure domain of the product design stage have been discussed. Additionally, knowledge use during design change management at this stage has also been covered.

2.1. *Change of functional requirements*

Change of functional requirements may have many reasons, for example, changes of customer demands, changes of government policies or changes of project aims for better competing with rivals (Rouibah and Caskey 2003, Clarkson *et al*. 2004). Changes occurring in functional requirements may affect three aspects of product design. (1) Functional requirement change needs to be verified according to customer requirements to make sure all the changes meet the original customer demands. As one of the most important inputs of a product design project, customer requirements should be monitored all the time during changes of functional requirements to make sure all the changes that meet the original customer demands. (2) Any change of a functional requirement may result in potential changes of other functional requirements depending on the interrelationships among them. These changes will be captured in the functional requirement model, so that causal impacts can be analysed and controlled. (3) Obviously, any changes in the functional requirement domain will affect physical structures that are correspondingly constructed according to functional requirements.

2.2. *Change of physical structures*

Change of physical structures is another important part of design change management. A lot of situations may give rise to changes in the physical structure design, for example, changes of functional requirement (discussed above), physical conflicts within solutions, solution changes on the supplier's side, technical innovation and manufacturing restrictions. Changes of physical structure may also directly affect three domains, namely the functional requirement domain, the physical structure domain itself and the manufacturing domain. In the functional requirement domain, any changes in the physical structure may change target outputs of related functions. Since one component is possibly involved in realisations of more than one function, the relationships between components and related functions need to be clarified. Therefore, any change in the physical domain needs be verified to make sure that target functional requirements have been met. In the physical structure domain itself, components are linked together by physical connections, which make it possible to realise demanded functions. Change of a component may potentially change operations of other components, which in turn may change the realisation of related functions. Figure 1 shows the change propagation routes within the functional domain and the physical structure domain and the routes between them. For example, change of the functional requirement F_n may require changes on components involved in the realisation of it, in this example let us say C_2. The change of C_2 could have influence on C_i via some behavioural or spatial relationships between them. C_i is one of the components involved in the realisation of F_i. Then C_i needs to be checked against functional requirement F_i to see if F_i can be satisfied. If not, then C_i or other components

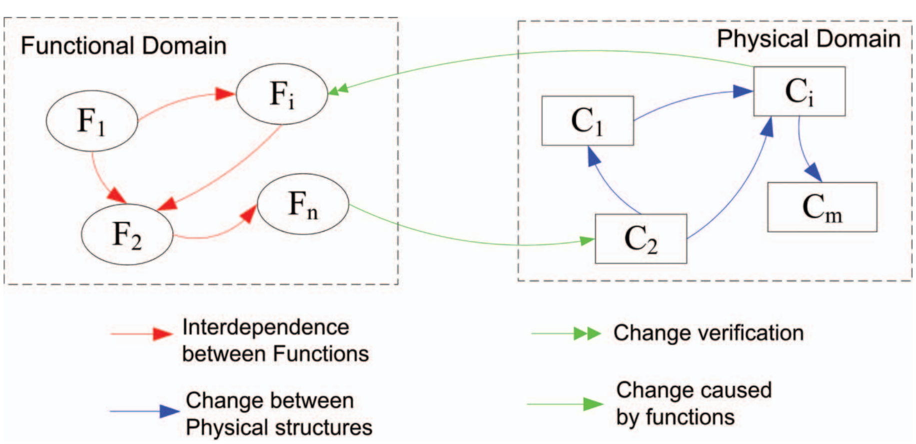


Figure 1. Design changes between functional and physical domains.

involved the realisation of F_i may need to be changed to accommodate the influence of the previous change. These components may be also involved in realisations of other functional requirements (for example F_1, F_2 in the diagram), so further changes may be needed. In this diagram, change of F_n directly cause changes of its components and also indirectly cause changes of other components and functional requirements. In the manufacturing domain, change of physical structure may change downstream activities, such as manufacturing process planning, SCM, risk and cost evaluation. Impacts on these product development activities need to be analysed or re-evaluated. Although change propagation between the functional domain and physical domain looks obvious from discussions in the last two sections, there is a lack of method for supporting tackling the change propagation process between the two domains, solving problems arising from the process and analysing change impacts. An important part of this article is focused on developing such a method to bridge the gap. The matrix-based method for analysing change propagations has been described in section 3.

2.3. *Conflict solving in design change management*

Conflict solving is one of the most concerned issues in design change management. In many cases, changes of a component or a function may require other parts of the design to change correspondingly. Furthermore, changes of these parts would cause changes of more parts of the design. This effect is the so called *change propagation or knock-on effect*. Actually, the reason why a change of a part of the design causes changes of other parts is because the initial change of the design may harm or obstruct operations of other components or satisfactions of other functional targets, which can be seen as functional or structural conflicts. In other words, change propagations are caused by design conflicts. Once there is no design conflict arising from any design changes, the change propagation stops. Although many design conflicts may have been solved during the change implementation by experienced engineers, many others may not be recognised in the design phase due to the lack of systematic methods and they would have been carried over to the manufacturing phase, which may cause a huge amount of cost in later phases. Therefore, an effectively analytical method is needed to identify conflicts in design change management and solve them as early as possible.

2.4. *Knowledge use in design change management*

Knowledge use is critical for design change management to ensure results from change analysis are fact based and consistent. There are mainly three aspects where design knowledge can be used to help solve change problems.

The first aspect is to identify design change modes. The design change mode is structured records of design changes implemented in previous applications. The point of having design change modes formalised in a knowledge repository is for frontline design engineers to find out whether similar design changes have happened. They can use these similar design change cases (if there is any) as references to help to find proper solutions and estimate potential change propagations and their impacts.

The second aspect is regarding design conflict solving. During a company's daily operations, there are a lot of solutions that have been proposed by engineers in attempt to tackle design conflicts arising from product development. These solutions whether successfully implemented or just on sketches are important assets of the company which should be properly generalised and deposited in the knowledge repository of the company. Once new design conflicts emerge and there is no similar design change mode that can be referred to, engineers can follow a formalised route to try to find proper solutions for them.

The third aspect is to use design knowledge to facilitate change impact analysis. Some knowledge of physical structure development has always been studied by companies, for example knowledge regarding developing time, developing cost and developing risks of solutions, components and parts. When a design change is initiated, engineers not only need to find its solutions and solutions for propagating changes, but also have to estimate the overall impact caused by the initial design change by taking consideration of time, costs and risks for development of new solutions. Therefore, decisions can be made for whether it is worth proceeding or not, or which parts of these solutions need to be modified, in order to make sure the change impact will not be too heavy to afford.

3. The proposed methodology

3.1. *Analysis of the DCM process*

There are three types of relationships existing in a product design, i.e. mapping relationship between functional requirements and physical structures, physical interaction relationship between structures, and spatial connection relationship between structures (Christophe *et al.* 2010). These relationships within product design largely cause change propagations. The method of design change management proposed in this article is based on analyses of these three types of relationships.

The process of design change management has been depicted in Figure 2. The whole process can be divided into three main steps: (i) system modelling of the product design; (ii) change propagation analysis based on the composite matrix and (iii) knowledge-based design conflict solving.

In the system modelling step, the product design with change (initial or propagated) applied has been modelled by three types of models, i.e. the functional structure model (in the form of SysML block definition diagram), the physical interaction model (in the form of SysML activity diagram) and the spatial connection model (in the form of CAD model). Each model clarifies one aspect of the product design correspondingly. By synthesising the relationships obtained from these three models, a composite matrix has been generated, which is critical to change propagation analysis. Differing from other matrix-based methods in some research in change propagation analysis, the composite matrix combines three types of relationships of the design together and provides an intuitive and dynamic way to capture change impacts across components and their functions. Explanation of each step is given in Section 3.3.

In the change propagation analysis step, the design change has been examined by checking

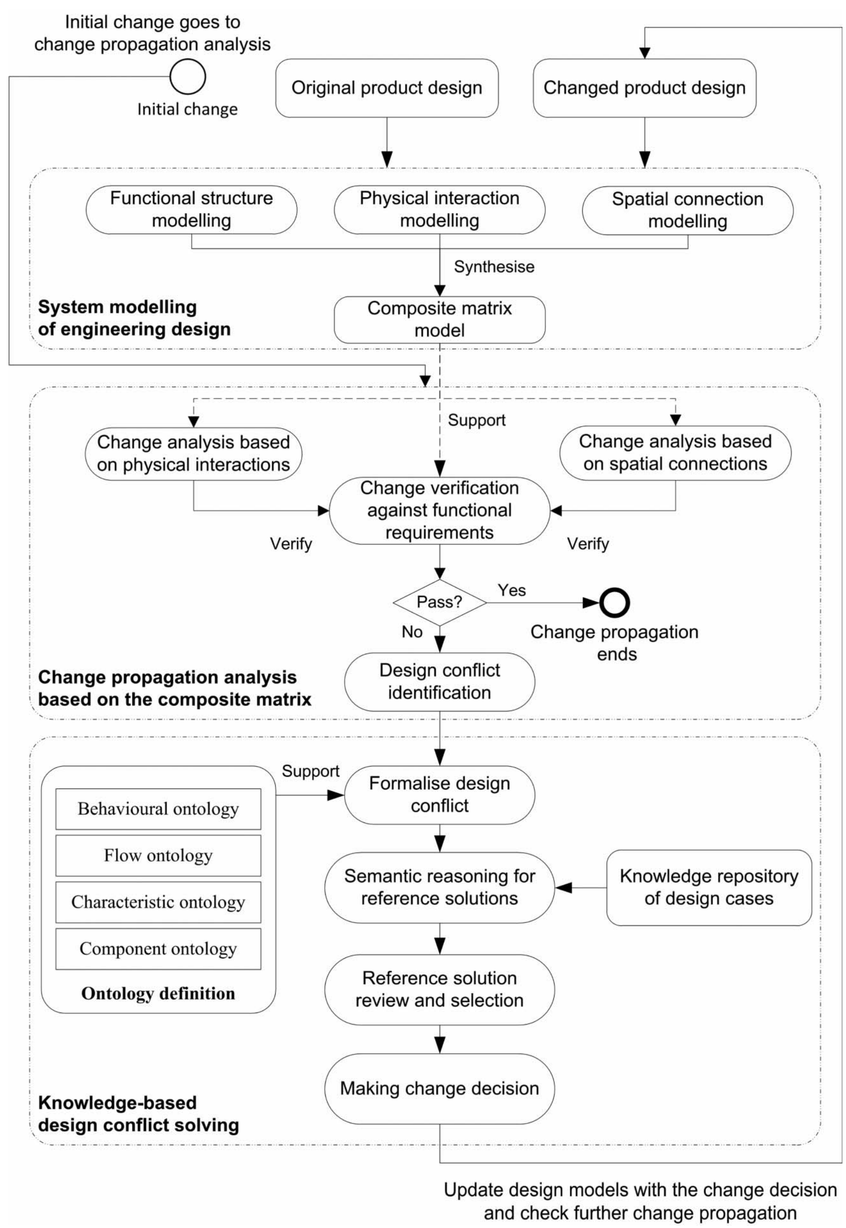


Figure 2. Analytical process of design change management.

changes to related flows (representing interactions between components) and spatial connections. Any components connected with these flows and spatial connections are then examined with functions that they serve for. If the effects on the components, which are caused by changes of those flows and spatial connections, make realisations of corresponding functions fail, it means there are design conflicts existing, which are caused by the design change. Design conflicts are then identified during the analysis with the composite matrix. The process of change propagation analysis using the composite matrix is described in detail in section 3.4.

In the knowledge-based design conflict solving step, design knowledge which is acquired from previous design cases is used to help solve design conflicts identified in the last step. Firstly, design conflicts are formalised by using pre-defined engineering ontology. Secondly, the formalised design conflicts are reasoned in the knowledge repository by semantically comparing with formalised general design cases stored in the knowledge repository. General solutions in the knowledge repository are formalised by the same set of the pre-defined engineering ontology. Thirdly, a prioritised list is generated with the most semantically similar general solutions at the top and the designers check those general solutions starting from the top of the list and their related design cases to find reference solutions for the design conflicts. At last, based on the reference solutions, proper solutions are worked out by designers and change decisions are made. The design with changes generated in this step will be re-modelled and further possible change propagation are analysed again as what is done in the last two steps.

3.2. *The industrial example used*

The industrial example used in this project to evaluate the proposed methodology is a cooling system, which is a critical part of a wind turbine (Figure 3). This new model of wind turbine is under development in our collaborative company that is a pioneer in gearless wind turbine development. Nearly 2000 wind turbines (by May 2010) based on their solutions have been deployed in Europe, Asia and America. There is a real design change scenario that the wind turbine needs to be deployed in a very sandy environment so that air filtering measure of the cooling system needs to be suppressed to prevent more sands than normal from coming to the cooling system and damaging it. This change causes some knock-on effects that give rise to changes on other parts of the system. The methodology proposed in this article is going to be used to solve this problem by modelling the cooling system, identifying design changes and related design conflicts and solving design conflicts by using a knowledge-based system.

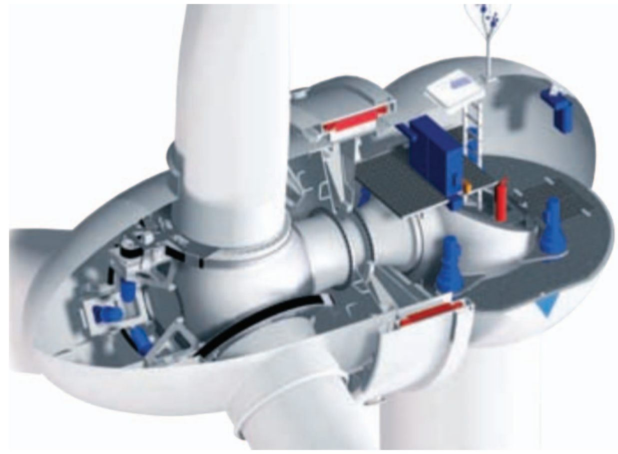


Figure 3. An inside view of a wind turbine.

3.3. *Modelling methods used in this project*

Modelling of engineering design includes three parts, namely functional structure model, physical interaction model and physical structure model. In this article, modelling methods for functional structure and physical interaction are adopted from SysML^{TM}, which is a comprehensive system engineering modelling language (Object Management Group 2008). The reason of using SysML is because it is a standard modelling method having intuitively visual presentations, standard descriptive language and software tool support. It can be easily understood by both human and computer, which is important for this project, since the methodology needs to be computerised to enhance its usability. For this reason, SysML is better than other modelling methods. Modelling of physical structure can be carried out by CAD systems, which will not be an emphasis in this article.

The functional structure is modelled by the block definition diagram (BDD) of SysML (Figure 4 depicts the functional structure of the cooling system). The BDD is used to model the hierarchically structural relationship of functions. It also helps to clarify the specifications of each function. The specification attribute of a function quantitatively or qualitatively represents what the function has to do, which is analysed by engineer from initial customer requirements or other requirements from various sources (e.g. technical restriction, management and government). Specifications are represented as attribute-value (could be precise value, value range or qualitative description) pairs. All of the sub-functions need to perform to meet their corresponding specifications so that specifications of their parent function can be met. Any changes to components need to be verified against its corresponding functional specifications to check whether these changes affect the realisation of functions. If functional specifications cannot be satisfied due to these changes, then other consequent changes need to be carried out.

Physical structure modelling carried out by CAD systems is intended to clarify the spatial relationship between components. When changes to a component happen, the spatial relationship helps designers find potential changes to neighbouring components based on their positions. The spatial relationship concerned in this model is all about static or kinematic connections between components, which is based on assembly relationships, but does not involve any flow-based physical interactions.

Physical interaction relationship is modelled by the internal block diagram to clarify the behavioural relationship between components. Figure 5 shows the interaction model of the cooling system. There are a variety of flows going through components, including energy flows, material flows and signal flows. A change on a component may cause changes of the flows going through it, which may also cause changes of upstream

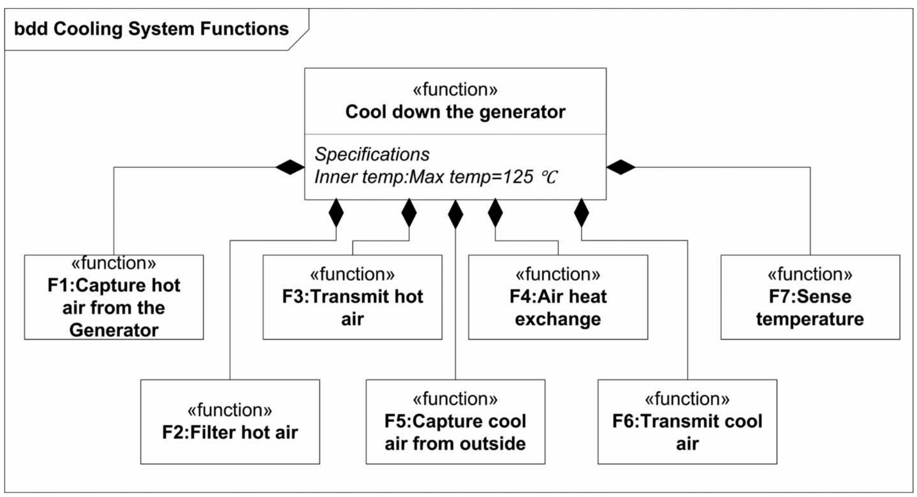


Figure 4. Functional analysis of cooling system.

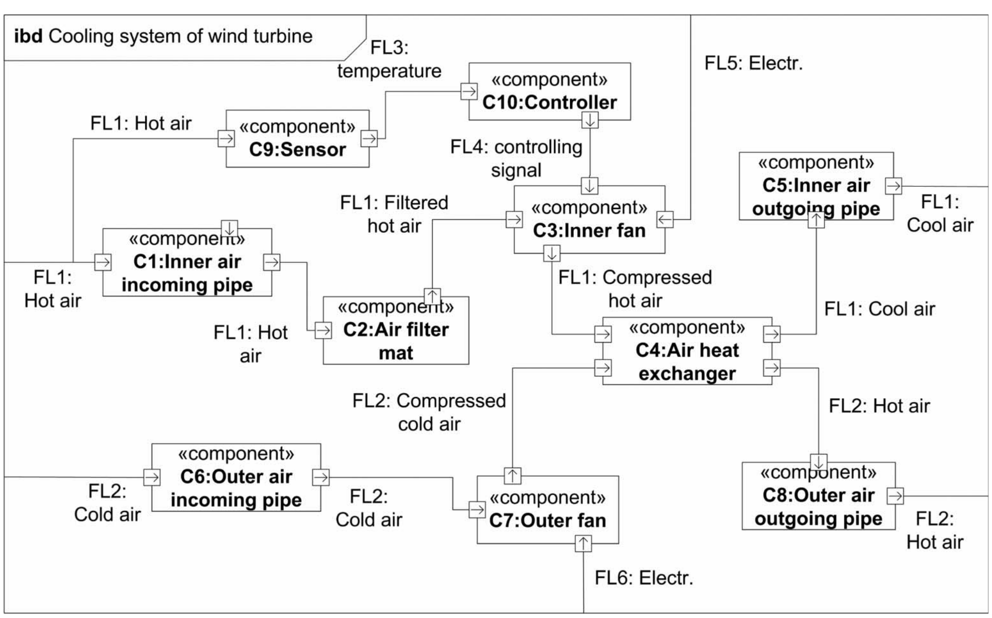


Figure 5. Analysis of interactions in the cooling system.

and downstream components involved in these flows. That is because the status changes of these flows may result in components not satisfying their corresponding functional requirements.

In order to be computerised, matrix analysis is employed to represent change propagations within and between functional requirement domain and physical structure domain. A composite matrix is constructed based on the results of modelling analyses of functional structure, physical interaction and physically spatial relationship (Figure 6). The matrix is composed of three parts that are marked by different colours. The first part (the green part) represents the mapping relationship between function and components. Elements in the first column represent physical components and elements in the green part of the first row represent functions. Each marked cell in the green part represents the involvement of a component in the realisation of a function. The second part (the blue part) represents the interaction relationship between components, which reflects the modelling results of physical interactions between components. Elements in the blue part of the first row represent flows in Figure 5. When there are changes happening on a component (an element in the first column), related flows going through it will be identified in the matrix. Components that these flows go through are also identified. The third part (the grey part) represents the physically spatial relationship between components. A marked cell in this part, the matrix means the component in the column is physically connected with the component in the row. Clarification of this type of relationship helps designers find potential propagating changes to neighbouring components. The next section presents further explanation of how potential propagating changes are identified.

3.4. *Identifying change propagation and design conflicts*

Change analysis is intended to uncover changes and their propagations by following connections within functional requirements and physical components and relationships between them. This section shows the idea of identifying change propagations and their impacts arising from a change of a component. The description of the method is associated with a scenario of improving air filtering as mentioned above and based on the composite matrix of change analysis (Figure 6). The process of identifying change propagation is described in the following steps. In this scenario, the change is triggered by a functional requirement called 'F2: Filter hot air'. Therefore, the analytical process starts from the function-component part of the matrix (the green part).

Step 1: Identify component changes caused by functional change. As mentioned above, because of the sandy environment where the wind turbine will be deployed, the current air filtering measure cannot meet the new functional requirement. In Figure 6, it shows components involved in the realisation of function, filter hot air (F2). In this case, there is just one component (C2, air filter mat) identified. To meet the sandy environment, the current air filter mat with a dust holding capacity 650 g/m^2 needs to be changed to a more effective one with dust holding capacity 750 g/m^2.

			Funtions							Flows						Components									
			F1: Capture hot air from the Generator	F2: Filter hot air	F3:Transmit hot air	F4: Air heat exchange	F5:Capture cool air from outside	F6: Transmit cool air	F7: Sense temperature	FL1: Air flow from the wind turbine to be	FL2: Air flow from outside for cooling	FL3: Temperature signal flow	FL4: Controlling signal flow	FL5: Electrical energy flow for the inner fan	FL6: Electrical energy flow for the outer fan	C1:Inner air incoming pipe	C2:Air filter mat	C3:Inner fan	C4:Air heat exchanger	C5:Inner air outgoing pipe	C6:Outer air incoming pipe	C7:Outer fan	C8:Outer air outgoing pipe	C9: Heat Sensor	C10:Controller
			wf(1)	wf(2)	wf(3)	wf(4)	wf(5)	wf(6)	wf(7)																
Components	C1:Inner air incoming pipe	wc(1)	x		x					x							x	x	x					x	
	C2:Air filter mat	wc(2)		x						x						x									
	C3:Inner fan	wc(3)	x		x					x			x	x		x				x					
	C4:Air heat exchanger	wc(4)				x				x	x						x				x				
	C5:Inner air outgoing pipe	wc(5)			x					x															
	C6:Outer air incoming pipe	wc(6)					x	x			x														
	C7:Outer fan	wc(7)					x	x			x				x					x	x				
	C8:Outer air outgoing pipe	wc(8)						x			x							x							
	C9: Heat Sensor	wc(9)	x						x			x													
	C10:Controller	wc(10)	x									x	x												

Figure 6. Composite matrix for change analysis.

Step 2: Identify potential affected components. The component changed in the above step may change the physical statuses of flows going through it and also it may change its neighbouring components due to changes of its spatial characteristics. Led by component C2, the row shows flows and neighbouring components that are potentially affected by the change of C2. In this case, flow FL1 (air from the generator) and neighbouring component C1 (inner air incoming pipe) are related to C1. The flow FL1 also goes through C1, C3, C4, C5, so these four components may also be potentially affected by the change of C1. The side effects of changing the air filter mat is that the mat with higher dust holding capacity is thicker and it causes larger air pressure drop, which can significantly reduce the efficiency of heat exchanging.

Step 3: Check change effects with related functions. Components that are affected by the flows and the spatial connections need to be checked whether the changed flows or the changed spatial connections would affect the realisations of their related functions. In this case, the air flow after the filter mat has a lower pressure, which means components C3, C4 and C5 would be potentially affected since the status of air through them is changed (see the column led by FL1). According to the analysis by engineers, the lower air pressure through C3 (inner fan) will weaken its performance. Also the lower air pressure through C4 (air heat exchanger) will reduce the efficiency of heat exchange. But it has almost no effect on C5 (the inner air outgoing pipe). The spatial change (thicker filter mat) has been considered as not noticeable change on C1 (inner air incoming pipe), since the change can be easily accommodated by the current design. Although in this case change caused by spatial connection is negligible, in many other cases, it may be significant and corresponding changes need to be made. For example, in this case, change the component C1 or add some other structures to accommodate the changes caused by spatial connections. Therefore, in this case, C3 and C4 have been identified as affected components which need to be changed to accommodate the previous change on C2.

Step 4: Identify and solve design conflicts. By analysing affected components, design conflicts can be identified. Taking C4 as an example, the changed input flow is the incoming air pressure that is lowered and the affected parameter is the heat exchange efficiency which is also lowered. The effect means that the heat exchange cannot meet the functional requirement F4. Therefore, this design conflict needs to be solved. In this article, we develop a knowledge-based method to help designers find reference solutions from previous design cases. Detailed discussion of how to solve design conflicts using a knowledge-based method is presented in section 3.3.

Step 5: Analyse change propagations caused by component changes in step 4. When a proper solution has been found in step 4, changes on affected components have been determined. These changes would potentially affect other components as well. In the above case, if component C4 has been changed flows FL1, FL2 and connected components C2, C6 may also be potentially affected. Thus, a next round of change analysis also needs to be carried out until there is no further change effect being identified, which means change propagation stops and change analysis initiated by the first change is finished.

4. Knowledge system support for design conflict solving

As the authors argued above, the reason why design changes propagate is that there are design conflicts between components when one or some of them changed. TRIZ, which is originated from Russia, is a set effective problem solving methods (Altshuller 1996). The contradiction matrix and invention principles are useful tools of TRIZ for solving technical conflicts. The idea of TRIZ to solve conflicts includes generally four steps: (1) identify technical conflicts; (2) generalise technical conflicts by using 39 engineering parameters; (3) find invention principles via a standard contradiction matrix and (4) explore specific solutions by following the indications of invention principles (Altshuller 1996, Fey and Rivin 2005). Although techniques of problem solving of TRIZ are innovative and inspirational to engineers, the method is difficult to master without a lot of trainings and long-time experience.

In this article, the authors proposed a knowledge-based method working in a similar way but more intuitive and easier to use. When a specific design conflict is identified during the change analysis, it will be formalised by referring to a set of predefined ontology. Then the formalised design conflict will be reasoned in the knowledge repository to find semantically similar generalised design cases. Therefore, specific design cases that are associated with generalised design cases can be retrieved. These selected design cases will be used as reference solutions for the current design conflict. Figure 7 depicts the process of how design conflicts have been solved.

4.1. Formalisation of design conflicts

4.1.1. Design conflict

Design conflicts are identified in the change propagation analysis based on the composite matrix.

Occurrence of design conflict has been simply depicted in Figure 8 in order to help understand the idea. Given that component 2 is one of the components serving a function, when there is a change request applied to a component (component 1), which has interactional connection with component 2, it may change the input flow of component 2, which may further affect its output flow. If the affected output flow cannot satisfy the requirement of the function, then it is said that there is design conflict occurring at component 2, which is caused by the previous change request to component 1.

4.1.2. Formalising design conflict with predefined ontology

Formalising a design conflict is actually not formalising the design conflict itself. In fact, it is about formalising the interactional model (called the meta-interactional model) of the component where the design conflict occurs. Figure 9 shows formalisation of a meta-interaction model.

A meta-interaction model includes a physical component where the conflict happens, the changed input flows and affected output flows. Both the input flows and the output flows are formalised by the flow ontology and characteristics ontology (ontology depicted in Figure 10). The flow ontology defines the type of the flow. The characteristics ontology defines the properties of the flow. For example, the gas flow normally has properties like pressure, temperature, moisture, etc. Properties formalised in this part should be critical to the operation of the component. Concepts of the characteristics ontology (a node of the ontology structure) are associated with concepts of the flow ontology in the form of their properties (shown in Figure 10).

The component is also formalised by the behaviour ontology and the component ontology. The behaviour ontology defines what the component does with the input flows and generates the output flows. The component ontology defines which type of component it is. The component ontology contains related component characteristics as its properties. These component characteristics are critical for the performance of the operation of the component. For example, in the cooling system, there are two characteristics of the heat exchanger that are important to its functionality. One is the area of the heat exchanger surface. Wider surface can have a higher heat exchanging efficiency. The other one is the material that the heat exchanger is made of.

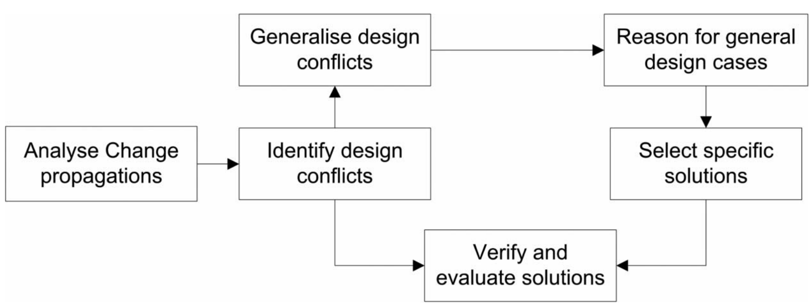


Figure 7. Process of solving design conflicts.

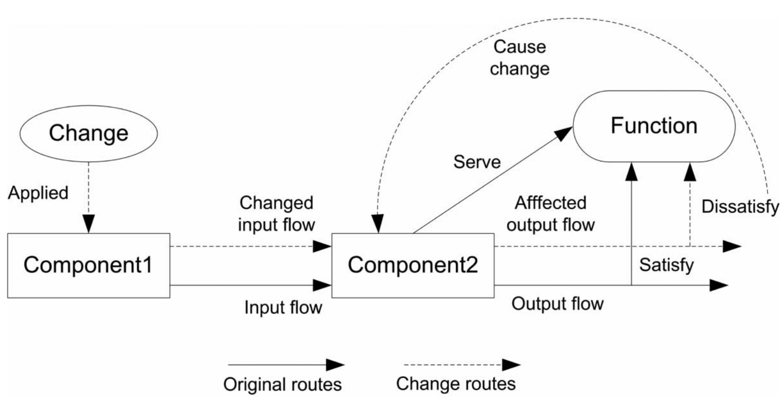


Figure 8. Design conflict occurring.

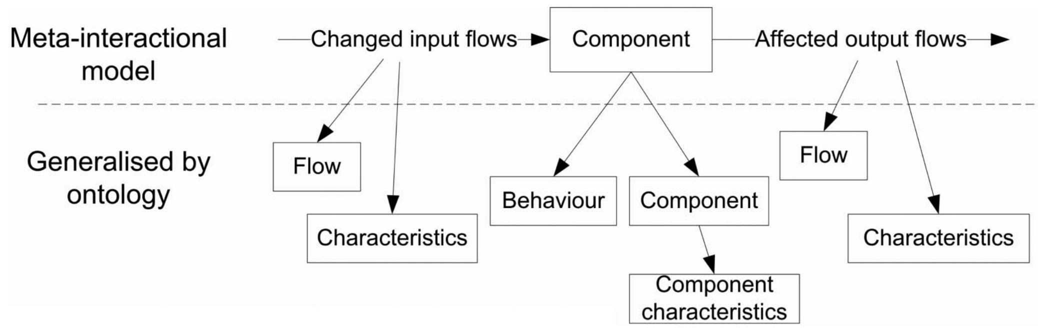


Figure 9. Formalisation of the meta-interaction model.

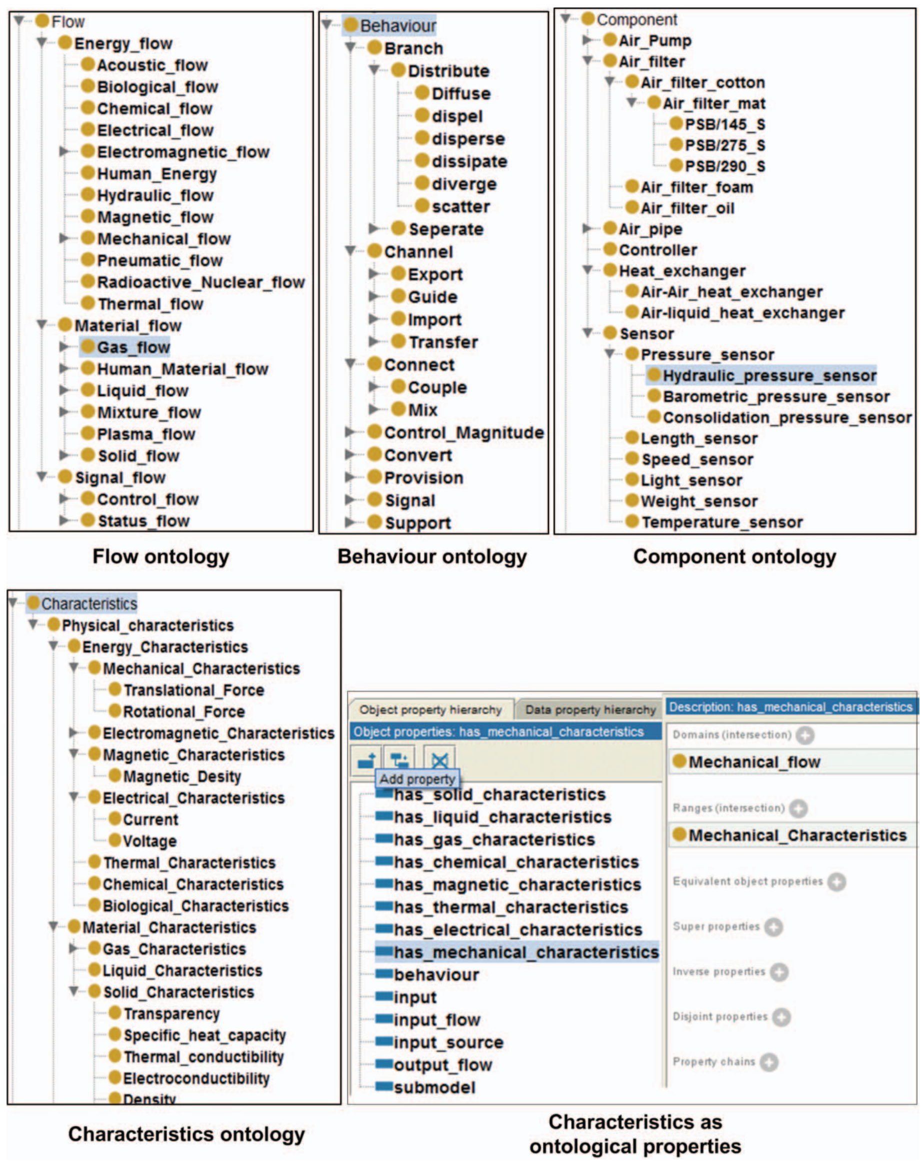


Figure 10. Ontology development for design change management.

Some material, for example bronze, has a better heat conduction performance than some others, for example steel.

The flow ontology and the behaviour ontology used in this article are adopted from the work of Hirtz *et al.* (2002). They proposed a functional ontology, so called functional basis, contains generalised functions (called as behaviour in this paper to differ from the term function of design) and flows which are seen as a useful and comprehensive engineering functional ontology by the authors. Also the authors developed domain-based component ontology and its characteristics by synthesising terms used in the investigated company. The characteristics ontology associated with flow ontology is also developed by synthesising related work by other researchers. We adopt Protégé as an ontology editor to formalise predefined ontologies in computer language. The tool which is developed by Stanford University is de facto in the academic area for ontology development.

Figure 10 shows some parts of the ontology, including definitions of flows, definitions of behaviours and definitions of components.

4.2. *The knowledge system for design conflict solving*

Figure 11 shows the framework of the knowledge base for design conflict solving. When a design conflict arises from design change analysis, it is formalised in the way as discussed in section 4.1. The formalised model of the design conflict will then be thrown into the knowledge system. The system will reason in the knowledge repository by analysing the semantic similarities between different concepts to find most similar generalised design cases. After that, design cases associated to these generalised design cases will be retrieved as reference solutions for the current design conflicts. Designers can adjust or adopt retrieved reference solutions to solve current problems. The method of generalising design cases is as the same as the way formalising meta-interaction model. It collects design cases and formalises their target problems or functions. The generalised design cases work as indices of those associated physical design cases.

In this system, design cases are collected from many sources including different functional departments and IT systems. The design cases are formalised in a hierarchical way, which clarifies functions or problems a design case is to address, the solutions used in this design case, the components involved in this solution and characteristics that contribute to the realisation of the function or the problem solving. The formalised structure is also been generalised by the predefined ontology.

4.3. *Reasoning method for design conflict resolving*

The reasoning method is critical to finding reference solutions for design conflicts arising from design change propagation analysis. The general approach to solving design conflicts is depicted in Figure 12.

In the reasoning method, one of the most important steps is to analyse semantic similarities between formalised design conflicts and generalised design cases stored in the knowledge repository. Given that a design conflict is defined as a definition (D) and all of the stored general design cases are defined as a set of definitions $\{D_1 \ldots D_n\}$, the first step is to find the most similar solutions from the set of general design cases. The algorithm of calculating the semantic similarities between a formalised design conflict and a generalised design case is comparing the semantic similarity of each corresponding element and then adding them up to get an overall semantic similarity.

Figure 13a represents hierarchically ontological definition of a group of concepts. The higher of levels a concept stays in, the more general semantic

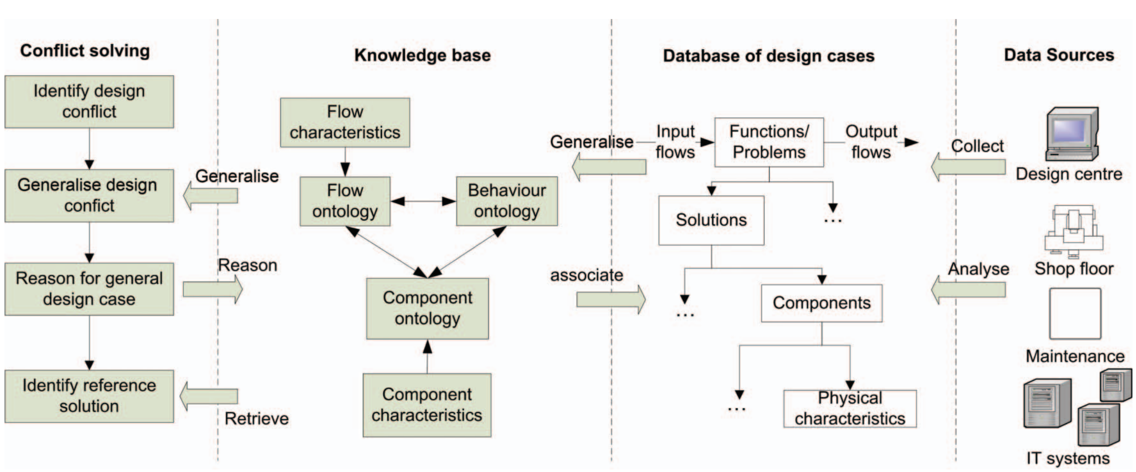


Figure 11. Framework of the knowledge system for conflict solving.

meaning it represents. While in lower levels, the semantic meaning of a concept is more specific. In Figure 13b, IC (S1) represents the semantic meaning of the concept S1. Since S1 is the parent of S11 and S12, S1 has a wider semantic meaning than S11 and S12, which means:

$$\mathrm{IC(S11)} \in \mathrm{IC(S1)}, \text{ and } \mathrm{IC(S12)} \in \mathrm{IC(S1)} \qquad (1)$$

Theoretically, the following equation can represent how S11 (a child) is semantically similar to S1 (a parent) by comparing scales of semantic meaning of each concept:

$$\mathrm{Sim(S11,S1)} = \mathrm{IC(S11)}/\mathrm{IC(S1)} \qquad (2)$$

While the similarity of S11 (a brother) to S12 (a brother) can be expressed as:

$$\mathrm{Sim(S11, S12)} = (\mathrm{IC(S11)} \cap \mathrm{IC(S12)})/\mathrm{IC(S12)} \qquad (3)$$

While in reality, it is well-known that exact semantic meaning of a concept is very difficult to measure. More often than not, people can tell the qualitative similarity between two concepts by their experience and common sense. Particularly, people in the professional area are better to tell similarities between concepts of professional terms some of which are not normally used by people out of the area. So in this research, a survey is developed and experienced engineers are interviewed to rate semantic similarities between child-concepts and their direct parent-concepts (e.g. Sim(S11, S1)) and also between their brother-concepts (e.g. Sim(S11, S12)). The survey and the rating process is a multi-criteria decision-making process, which is not discussed in details in this article.

With rated similarities between parent–child concepts and brother concepts, the similarity of any two concepts (e.g. S111 and S22) can be expressed as:

$$\mathrm{Sim(S111,\ S22)} = \left\{ \mathrm{Sim(S111,\ S11)} \times \mathrm{Sim(S11,\ S1)} \times \mathrm{Sim(S1,\ S2)} \times \mathrm{Sim(S22,\ S2)} \right\} \qquad (4)$$

Based on the ideas of rating and calculating similarities between concepts, the similarity between a

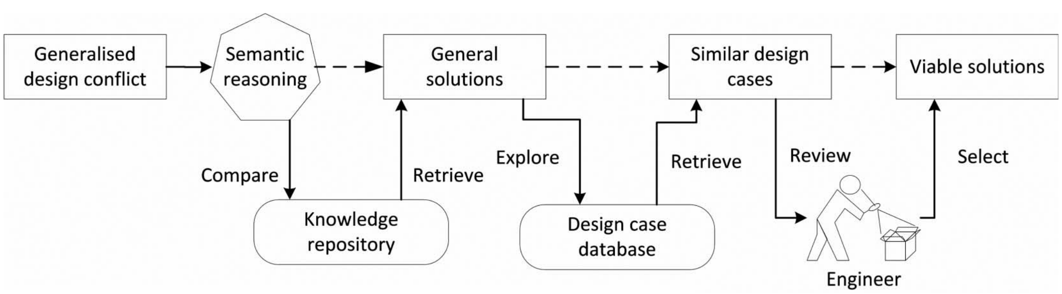


Figure 12. Reasoning approach to design conflict solving.

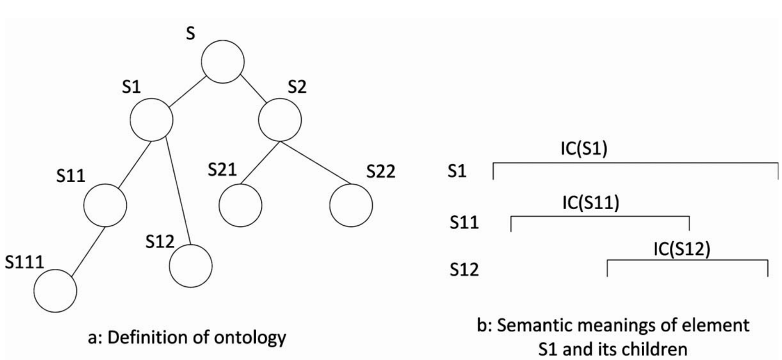


Figure 13. Comparison of semantic meanings between concepts.

formalised design conflict and a generalised design case can be described as:

$$\mathrm{Sim}\,(\mathrm{DC},\mathrm{GD}) = \Big\{\mathrm{Sim}\,(\mathrm{CF_{DC}},\mathrm{CF_{GD}}) \times \mathrm{Sim}\,(\mathrm{Behaviour_{DC}},\mathrm{Behaviour_{GD}}) \times \mathrm{Sim}\,(\mathrm{Component_{DC}},\mathrm{Component_{GD}}) \times \mathrm{Sim}\,(\mathrm{AF_{DC}},\mathrm{AF_{GD}})\Big\} \quad (5)$$

Where DC represents design conflict, GD represents generalised design case, CF represents changed flow and AF represents affected flow. Similarity between changed flows can be represented as:

$$\mathrm{Sim}\,(\mathrm{CF_{DC}},\mathrm{CF_{GD}}) = \mathrm{Sim}\,(\mathrm{Flow_{DC}},\mathrm{Flow_{GD}}) \times \mathrm{Sim}\,(\mathrm{Character_{DC}},\mathrm{Character_{GD}}) \quad (6)$$

Similarity between affected flows can be represented as:

$$\mathrm{Sim}\,(\mathrm{AF_{DC}},\mathrm{AF_{GD}}) = \mathrm{Sim}\,(\mathrm{Flow_{DC}},\mathrm{Flow_{GD}}) \times \mathrm{Sim}\,(\mathrm{Character_{DC}},\mathrm{Character_{GD}}) \quad (7)$$

By comparing the overall similarities between the formalised deign conflict and the generalised design case, a set of prioritised similarity values are generated:

$$\{\mathrm{Sim}(\mathrm{DC},\ \mathrm{GD_1}), \mathrm{Sim}(\mathrm{DC},\mathrm{GD_2}), \ldots, \mathrm{Sim}(\mathrm{DC},\mathrm{GD_n})\}$$

By exploring and reviewing design cases associated with retrieved generalised design cases from higher priority to lower priority, the suitable design cases are chosen as reference solutions for the target design conflict.

4.4. Results of the industrial example and system implementation

The example used in previous sections has been processed in this system. Application of the methodology has been partially displayed during the discussion of the methodology. In order to have an overview of the results and the general process in an intuitive way, a tabular form has been used to organise the results (shown in Figure 14).

A three-tier web-based pilot system has also been developed to implement the proposed methodology for engineering design change management. The pilot system is partly developed. Integrations with other business systems like PDM, ERP and SCM are going to be done at the next stage of this research. Software systems involved in this prototype include MySQL as an infrastructure of data storage, Tomcat 6.0 as a web server and servlet container, Java enterprise edition (Java EE) as the business implementation architecture and also JavaServer Page as a presentation technology. Figure 15 shows an overview interface of design conflict solving of design change analysis. The page shows the final stage of the design conflict solving analysis process, which is trying to get reference solutions from the knowledge repository to help designers work out a viable solution for the current design conflicts.

5. Conclusions and further work

Any design changes either in functional requirement domain or in the physical structure domain will potentially affect operations of other parts. Change propagations and their impacts are difficult to be captured, which makes product design in uncertainty. The authors argue that design change propagation is caused by design conflicts arising from the initial change and changes afterwards. Change propagations are difficult to predict without knowing their preceding change solutions, since all following changes are based on the solutions of preceding changes. Knowledge from previous design change cases is an important asset for companies. Many design conflicts arising from change analysis can be tackled by reusing well-formalised and -managed knowledge abstracted from previous design cases. In this article, a methodology for design change management has been proposed to implement these ideas by using modelling method, matrix analytical method and knowledge support. First, the system engineering modelling method captures critical interactions between functions and components. The matrix-based analytical method helps to trace change propagations and identify design conflicts by following mapping relationship between functions and components, and behavioural and spatial connections between components. By using the design conflicts formalisation approach, design conflicts can be formalised based on predefined ontology, which makes knowledge reasoning in the knowledge base possible. The framework of the knowledge base is proposed to collect knowledge of design change from previous design cases. With help of the knowledge base and the reasoning method, designers are able to find proper solutions and evaluate their potential impacts. A prototype system

Design conflict description			Formalised design conflict				General solutions	Design cases
Change	Cause	Effect	Formalisation		Generalisation by Ontology			Design documents
Change Air Filter Mat	Lower air pressure (FL1) from the inner air incoming pipe (C1)	Reduce the efficiency of heat exchanging (C4)	**Affected Function**	Air heat exchange (F4)	**Behaviour:**	Conduct	1, Change the material of the conduction part; 2, Increase the area of conduction interface; 3, Improve the structure of air vessel; 4, Increase the velocity of air flow	1, Passive cooling system of the unclear reactor (File no: ****) 2, Patent: DE 19636591C; 3, Central air conditioner (File no. *****);
					Object:	Thermal flow, Gas flow		
			Changed input flow	Air flow from incoming pipe(FL1) with lowered pressure	**Flow:**	Thermal flow, Gas flow		
					Characteristics	Pressure		
					Change:	Reduced		
			Affected output flow	External hot air from the exchanger (FL2) with less heat exchanged	**Flow:**	Thermal flow, Gas flow		
					Characteristics	Temperature		
					Change:	Increaced		
			Affected Component	Air Heat Exchange (C4)	**Component**	Air-Air heat exchanger		
					Characteristics	Material, contact area		

Figure 14. Results of the example processed in the knowledge system.

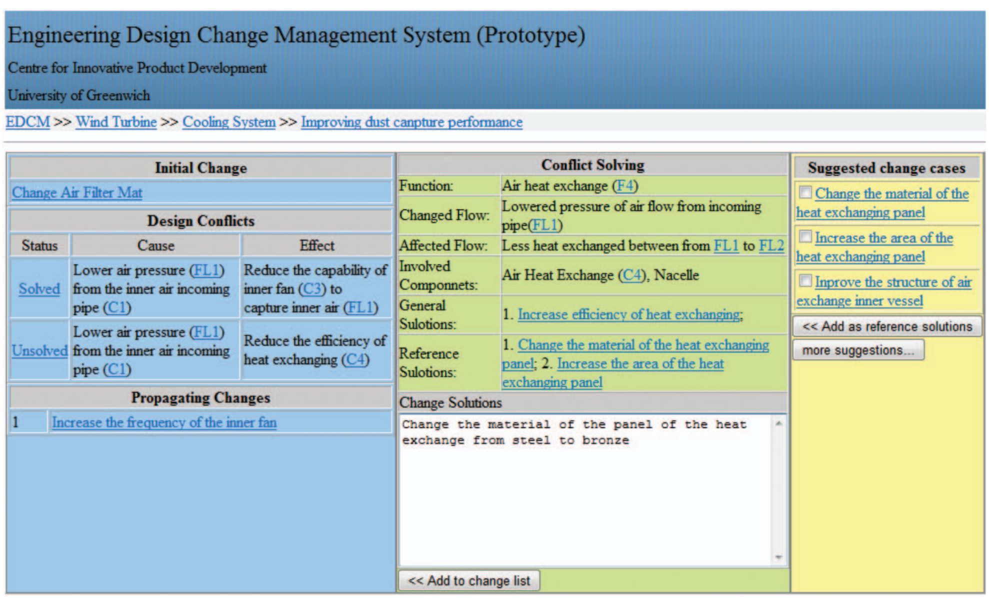


Figure 15. Prototype system implementation of CDM.

has been developed to show that the idea of this article is technically feasible and can benefit companies with IT technologies.

The emphasis of further work will be put on development of methods for evaluating design change impact in terms of factors of project success. Any changes made in product development need to be evaluated in terms of development time, development risk and development cost. However, in practice, impacts of design changes are normally estimated by experienced engineers or other staff in the company. The results of change impact evaluations could be very inconsistent with estimations from different people who have different experience and different knowledge backgrounds. In the next stage of this research, an approach to integrating with other enterprise systems will be developed. This approach is to acquire knowledge regarding contributions that previous products make to related projects in terms of factors of project success. With help of knowledge acquired, an algorithm will be developed to calculate change impact more accurately.

Acknowledgements

This project has been supported by the wind turbine design company Vensys Energy AG in Germany and the leading wind turbine manufacturer Goldwind Ltd. in China. Some of their engineering designers provided advice in developing the methodology and the illustrative example.

References

Altshuller, G.S., 1996. *And suddenly the inventor appeared: TRIZ, the theory of inventive problem solving*. 2nd ed. Worcester, MA: Technical Innovation Center.

Ariyo, O.O., Eckert, C.M., and Clarkson, P.J., 2009. Challenges in identifying the knock-on effects of engineering change. *International Journal of Design Engineering*, 2, 414–431.

Balogun, J. and Jenkins, M., 2003. Re-conceiving change management: a knowledge-based perspective. *European Management Journal*, 21 (2), 247–257.

Christophe, F., Bernard, A., and Coatanéa, É., 2010. RFBS: a model for knowledge representation of conceptual design. *CIRP Annals-Manufacturing Technology*, 59 (1), 155–158.

Clarkson, J. and Eckert, C., 2005. *Design process improvement: a review of current practice*. London: Springer.

Clarkson, P.J., Simons, C., and Eckert, C., 2004. Predicting change propagation in complex design. *Journal of Mechanical Design*, 126 (5), 788–797.

Do, N., Choi, I.J., and Song, M., 2008. Propagation of engineering changes to multiple product data views using history of product structure changes. *International Journal of Computer Integrated Manufacturing*, 21 (1), 19–32.

Eckert, C., Clarkson, P.J., and Zanker, W., 2004. Change and customisation in complex engineering domains. *Research in Engineering Design*, 15 (1), 1–21.

Eckert, C.M., *et al*., 2006. Supporting change processes in design: complexity, prediction and reliability. *Reliability Engineering & System Safety*, 91 (12), 1521–1534.

Fey, V. and Rivin, E.I., 2005. *Innovation on demand: new product development using TRIZ*. Cambridge: Cambridge University Press.

Hirtz, J., *et al*., 2002. A functional basis for engineering design: reconciling and evolving previous efforts. *Research in Engineering Design*, 13 (2), 65–82.

Huang, G.Q., Yee, W.Y., and Mak, K.L., 2001. Development of a web-based system for engineering change management. *Robotics and Computer-Integrated Manufacturing*, 17, 255–267.

Huang, G.Q., Yee, W.Y., and Mak, K.L., 2003. Current practice of engineering change management in Hong Kong manufacturing industries. *Journal of Materials Processing Technology*, 139 (1–3), 481–487.

Jarratt, T.A.W., *et al*., 2003. Environmental legislation as a driver of design. *In*: A. Folkeson, K. Gralen, M. Norell, and U. Sellgren, eds. *The 14th international conference on engineering design (ICED'03)*, 19–21 August 2003 Stockholm, Sweden.

Keese, D.A., Seepersad, C.C., and Wood, K.L., 2009. Product flexibility measurement with enhanced change modes and effects analysis CMEA. *International Journal of Mass Customisation*, 3, 115–145.

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

Kocar, V. and Akgunduz, A., 2010. ADVICE: a virtual environment for engineering change management. *Computers in Industry*, 61 (1), 15–28.

Lee, H., *et al*., 2006. Capturing and reusing knowledge in engineering change management: a case of automobile development. *Information Systems Frontiers*, 8 (5), 375–394.

Mckay, K., *et al*., 2003. Design change impact analysis during early design specification. *International Journal of Computer Integrated Manufacturing*, 16 (7), 598–604.

Object Management Group, 2008. *OMG Systems Modeling Language (OMG SysML) V1.1*. [online]. Object Management Group. Available from: http://www.omg.org/spec/SysML/1.1/PDF [Accessed 16 May 2009].

Ouertani, M.Z., 2008. Supporting conflict management in collaborative design: an approach to assess engineering change impacts. *Computers in Industry*, 59 (9), 882–893.

Ouertani, M.Z. and Gzara, L., 2008. Tracking product specification dependencies in collaborative design for conflict management. *Computer-Aided Design*, 40 (7), 828–837.

Palani Rajan, P.K., *et al*., 2005. An empirical foundation for product flexibility. *Design Studies*, 26 (4), 405–438.

Pikosz, P. and Malmqvist, J., 1998. A comparative study of engineering change management in three Swedish enginnering companies. *In*: H. Lipkin and F. Mistree, eds. *Design engineering technical conferences and computers in engineering conference*, 13–16 September 1998 Atlanta, Georgia.

Rouibah, K. and Caskey, K.R., 2003. Change management in concurrent engineering from a parameter perspective. *Computers in Industry*, 50 (1), 15–34.

Shiau, J.-Y. and Wee, H.M., 2008. A distributed change control workflow for collaborative design network. *Computers in Industry*, 59 (2–3), 119–127.

Suh, N.P., 2001. *Axiomatic design: advances and applications*. New York: Oxford University Press.

Wright, I.C., 1997. A review of research into engineering change management: implications for product design. *Design Studies*, 18 (1), 33–42.

A Methodology for Engineering Design Change Management Using Modelling and Problem Solving Techniques

G. Fei[1], J. Gao[1], O.O. Owodunni[1], X.Q. Tang[2]
[1] Centre for Innovative Product Development, School of Engineering, the University of Greenwich, ME4 4TB, UK
[2] School of Mechanical Engineering and Automation, Beihang University, Beijing, 100083, China

Abstract: The importance of design change management in engineering product development has been widely reported. However, most research focuses on the engineering change in the manufacturing phase, the later phase of product development. In this paper, a reference model for design change management has been proposed. A modelling method is employed to enhance the traceability of changes occurring between the functional model and the structural model. A matrix based method has been developed to capture change propagations at the structural level and the parametric level of product design. A structural method of change propagation analysis and change impact evaluation has been developed. Finally, conclusions are drawn and scope for further work of this research is indicated.

Keywords: engineering design, change management, product modelling, problem solving, Triz, SysML

1. Introduction

It is recognised that the design stage of product development can determine the largest part of the costs occurring in a product's life cycle. It is critical to keep the early product design consistent and keep all the changes at this stage under control. The importance of design change management (DCM) in engineering product development has been generally reviewed and most research focuses on the engineering change occurring in the manufacturing phase, the later stage of product development [1].

From literature reviewed so far, changes of physical structures have been considered by many researchers. However, most of them focused on propagation changes in the manufacturing phase caused by changes of physical structure, by changes in methods of organisational configurations and working processes [2], and implement change management in computer aided management [3,4]. In this paper, a method for design change management is proposed by putting the emphasis on an analysis of changes between the functional requirement domain and the physical requirement domain. A modelling method is employed to enhance the traceability of changes occurring between the functional model and the structural model. A matrix based method has been developed to capture change propagations at structural level and the parametric level of product design. A structural method of change propagation analysis and change impact evaluation has been developed.

The research scope of design change management (D CM) is restricted in dealing with changes occurring between functional requirements and physical structures. They are mainly three aspects: (1) capture changes and their propagations between functional requirements and physical structures; (2) identify change modes and solve conflicts arising from design changes; (3) analyse impacts of changes in terms of manufacturing process, developing cost, developing risk. In this paper, we mainly focus on the first aspect

Change of functional requirement may have many reasons, for example, customer demand change, government policy change, or project aim change for competing better with rivals, and so on. Changes occurring in functional requirements may directly affect three aspects of product design. (1) Functional requirement change needs to be verified according to customer requirements to make sure all the changes meet the customer's original demands. (2) Any change of a functional requirement may have potential changes of other functional requirements depending on the interrelationships between them. These changes will be captured in the functional requirement model so that causal impacts can be analysed and controlled. (3) Obviously, any changes in the functional requirement domain will affect physical structures which are correspondingly constructed according to functional requirements.

Change of physical structure is another important part in this research. A lot of situations may give rise to change in the physical structure design, for example, change of functional requirement (discussed above), physical conflict within the solution, solution change on the supplier's side, technical innovation, manufacturing restriction, and so on. Changes of physical structure may also directly affect three domains, namely the functional requirement domain, the physical structure domain itself and the manufacturing domain. In the functional requirement domain, any changes in the physical structure may change target outputs of related functions. Since one component is possibly involved in realisations of more than one function, the relationships between components and related functions need to be clarified. In the physical structure domain itself, components are linked together by some physical connections, which make it possible to realise some desired functions. Change of a component may potentially change operations of other components, which in turn may change the realisation of related functions. In the manufacturing domain, change of physical structure may change downstream developing activities, such as manufacturing process planning, risk and cost evaluation. Impacts on these product developing activities need to be analysed or re-evaluated.

Conflict solving is one of the issues of most concern in design change management. In many situations, changes of a component or a function may just cause other parts of the design to change correspondingly. However, in some cases, changes may cause functional or physical conflicts. That means they may harm or obstruct operations of other components or realisations of other functions. Some conflicts may not have been recognised in the design phase and have been carried over to the manufacturing phase which may cause a huge amount of cost in later phases. Therefore, an effectively analytical method is needed to identify conflicts in the design phase and solve them as early as possible.

2. The process of design change management

A structural working process would be helpful for designers to achieve their objectives. A diagram of the proposed design change management process is shown in Fig 1. It has four main sections (not shown explicitly).. In the first section (top rectangles), dependent relationships among functional requirements and interacting relationships among physical components are clarified. In the second section, the functional model and the physical structure model of engineering products are constructed. Mapping connections between the functional requirement domain and the physical structure domain are also identified. Therefore, designers can be always aware of which interaction in the physical structure is contributing to a realisation of which part of a function. In the third section, changes either in functional requirements or in the physical structures are analysed. Propagations of each design change are identified by following functional dependency relationships and physical interacting relationships. After that, a change propagation matrix is constructed. Weighed values are are assigned in the matrix to differentiate the relative importance of each change in terms of changing impacts in functional requirement domain and physical structure domain. Change impacts are then calculated. In the last section, knowledge based methods are used to solve design conflicts by integrating TRIZ. TRIZ is able to guide designers to find proper solutions for design conflicts via its design principles. By integrating TRIZ with domain knowledge, conflict solving for design change will be more effective and efficient. Due to the page limit, the last section will not be discussed in detail. However, solving conflicts from design changes will be the emphasis of the research work at the next stage.

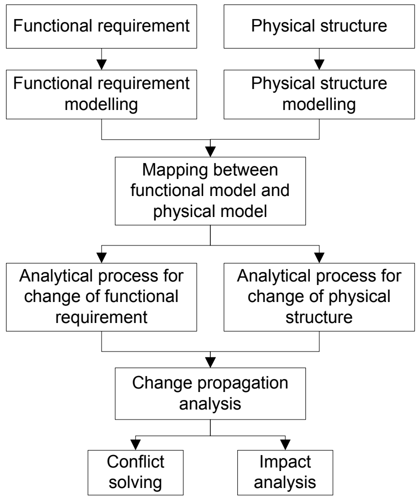


Fig. 1. The process of design change management

3. Analytical method for design change management – an illustrative example

In this section, analytical methods for design change management are introduced and demonstrated with industrial examples from our collaborator. Techniques for analysing design changes include modelling methods for functional requirements and physical structures, a method for constructing matricies for tracing change propagation, and parametric analysis of impact factors of design changes. The industrial example demonstrates a cooling system of a wind turbine which is under development in the collaborative wind turbine design company.

Modelling methods for functional requirements and physical structures are adopted from SysML™ which is a comprehensive system engineering modelling language [5]. The activity diagram of SysML is designed to capture interactions (behaviours referring to SysML) among functions. Each function can be treated as an action which is fed with functional parameters from some actions (functions) and also outputs functional parameters to other actions (functions) as well (an example is shown in Fig. 2). Thus in this paper, when we mention action in terms of functional requirements, we mean functions. The internal block diagram (IBD) captures material, energy and information flows coming in and going out of a physical structure (example depicted in Fig. 3). Every block in the IDB represents a component (subassembly) of a product at a certain level of detail. Flows attached to a block mean its physical interactions with other components. Functional interactions and physical interactions captured in modelling methods can reflect possible propagation paths of changes.

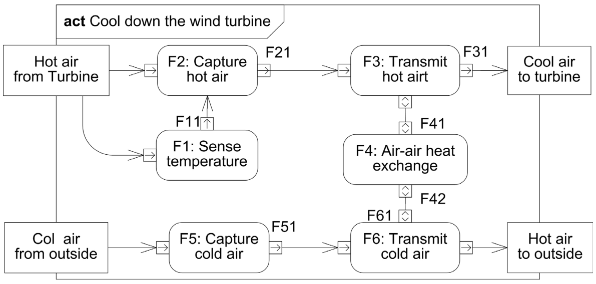


Fig. 2. Activity diagram of a cooling system

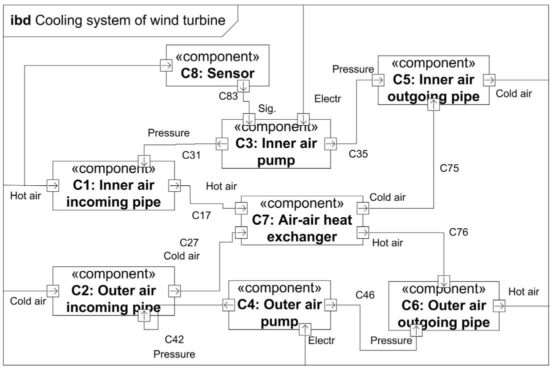


Fig. 3. Internal block diagram of a cooling system

In order to be computer processable, matrix analysis is employed to represent change propagations within and between a functional requirements domain and a physical structure domain. Construction of matrices is based on the results of modelling analyses of activity diagrams and internal block diagrams (depicted in Fig. 4 (a) (b) (c)). In the Fig. 4, propagation matrix (a) represents interactions between functional requirements; propagation matrix (c) represents interactions between physical structures; and propagation matrix (b) represents relationships between functional requirements and physical structures. Basically, activity diagrams and internal block diagrams are directed graphs. Therefore, if there is a flow from function F1 to function F2, the cell (F1, F2) in matrix (a) will be marked to represent this interaction, which is the same in matrix (c). In matrix (b), if a component (e.g. C1) is involved in realising a function (e.g. F2), then the corresponding cell will be marked (e.g. (C1, F2)). The matrix (b) represents relationships between the functional requirement domain and the physical structure domain. It clarifies which physical interactions would influence which functional flows.

	F1	F2	F3	F4	F5	F6
F1		X				
F2			X	X		
F3						
F4			X			X
F5						X
F6				X		

(a)

	F1	F2	F3	F4	F5	F6
C1		X	X			
C2					X	X
C3		X				
C4					X	
C5			X			
C6						X
C7				X		
C8	X					

(b)

	C1	C2	C3	C4	C5	C6	C7	C8
C1							X	
C2							X	
C3	X				X			
C4		X				X		
C5								
C6								
C7					X	X		
C8			X					

(c)

Fig. 4. Matrices of interaction analysis

Depending on different levels of detail at different design stages, a change analysis can be carried out at two levels, i.e. the structural level and the parametric level.

Change analysis at the structural level

Change analysis at the structural level is intended to uncover changes and their propagations by following connections within functional requirements and physical components and relationships between them, where parametric connections are not considered or disclosed. Changes at this level mean changes of an entity, for example a function or a component, but not a part of the entity, for example a functional parameter or a physical parameter. The idea of identifying change propagations and their impacts arising from a change of an entity is described in Fig. 5 (ignore broken line boxes for now).

The relative importance of an entity can be found by comparing entities within a same domain at the same level of detail. The relative importance of a function f in the functional requirement domain can expressed as *FM*(*f*). It is calculated by comparing other functions at the same level of detail, in terms of their contributions to the product functionality (1: indifferent; 3: fairly important; 5: very important). The relative importance of a component *c* in the physical structure domain can be expressed as *CM*(*c*). Differing from functional importance, relative importance of a component is obtained by its contributions to related functions and importance of related functions. For example, if *C1* in Fig. 4 (b) is related with *F2* and *F3*, then

$$CM\ (c_1) = CFM\ (c_1, f_2) \times FM\ (f_2) + CFM\ (c_1, f_3) \times FM\ (f_3) \qquad (1)$$

where *CFM*(*c*, *f*) represents the contribution of component *C* to the realisation of function *F*.

Interactive relationships can be found from change propagation matrices. Matrices (a), (b), (c) in Fig.4 respectively show the interactive relationships between functions (expressed as *FFM*(*f, f'*)), the interactive relationships between functions and components (expressed as *CFM*(*c, f*)), and interactive relationships between components (expressed as *CCM*(*c, c'*)). Importance of interactive relationships at this level are qualified by number 0 (no influence), 1(indifferent), 3 (fairly strong), 5 (very strong), which represent how strongly an entity influences another. Therefore, change impact between two functions (*f*, *f'*) can be calculated as:

$$CI(f,f')=FM(f)\times FFM(f,f')\times FM(f') \quad (2)$$

Change impact between two components (c and c') can be calculated as:

$$CI(c,c')=CM(c)\times CCM(c,c')\times CM(c') \quad (3)$$

Change impact between a function (f) and a component (c) can be calculated as:

$$CI(f,c)=FM(f)\times CFM(f,c)\times CM(c) \quad (4)$$

Fig. 5 describes the change analysis process at the structural change level (not including broken lined blocks).

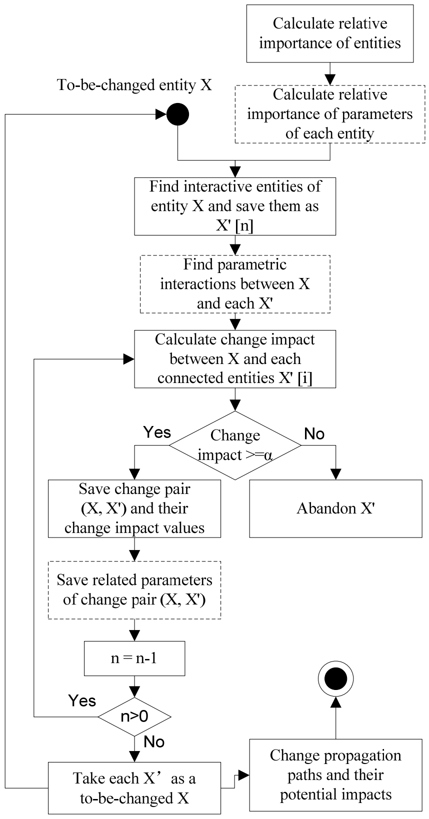


Fig. 5. Change propagations and impacts identification

Change analysis at the parametric level

Change analysis at the parametric level is to find change propagations and evaluate their impacts in the perspective of parametric changes. It has a similar analytical process as the change analysis at the structural level. The differences are that interactive relationships are between parametric flows instead of entities. Interactions of parametric flows are considered in the impacts evaluation of changes.

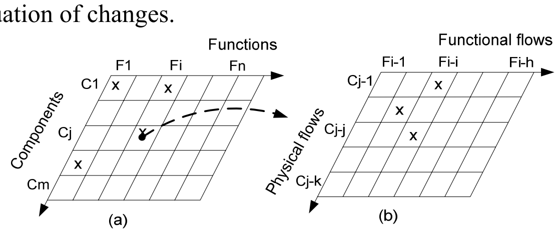


Fig. 6. Interaction analysis of parametric flows

As depicted in Fig. 6, the connections between function Fi and component Ci in Fig. 6(a) are composed of parametric connections shown in Fig. 6(b). Therefore, say if a designer wants to change the parameter Cjj of component Cj, the change impact between Cj and function Fi can be expressed as

$$PCI(c_{jj},f_{ii})=FM(f_i)\times(PCM(c_{jj})\times PCFM(c_{jj},f_{ii})\times PFM(f_{ii}))\times CM(c_i) \quad (5)$$

where $PCM(c_{jj})$ represents the relative importance of parameter c_{jj} in component C_j, $PFM(f_{ii})$ means the relative importance of parameter f_{ii} in function F_i, and $PCFM(c_{jj}, f_{ii})$ means importance of contribution from parameter c_{jj} to functional parameter f_{ii}. The analytical process is depicted in Fig. 5 including broken line boxes.

4. Conclusions and Further Work

Any design changes either in functional requirement domain or the physical structure domain will potentially affect operations of other parts. Change propagations and their impacts are difficult to be captured, which makes from uncertainty in product design. A method for design change management has been proposed to tackle this problem by using a modelling method and a specific analytical method. The modelling method captures all the interactions between functions and components. The analytical method helps to trace change propagations and evaluate their potential impacts.

Further work includes synthesising change modes in product design, analysing their change impacts, developing a knowledge based conflict identifying and solving method by integrating method, and a knowledge based technique for non-functional impact evaluation by considering developing process change, developing risk change, and developing cost change. Domain knowledge will be investigated and integrated with TRIZ principles to help designers find solutions for changes and conflicts.

References

[1] Wright IC. 1997. A review of research into engineering change management: implications for product design. Des Stud 18:33–42

[2] Rouibah, K. & Caskey, K. R. 2003. Change management in concurrent engineering from a parameter perspective. Computers in Industry, 50, 15–34.

[3] Pikosz, P. & Malmqvist, J. 1998. A comparative study of engineering change management in three Swedish engineering companies. Proceedings of DETC'98. Atlanta, Georgia.

[4] Huang, G. Q., Yee, W. Y. & Mak, K. L. 2001. Development of a web-based system for engineering change management. Robotics and Computer-Integrated Manufacturing, 17, 255–267.

[5] OBJECT MANAGEMENT GROUP. 2008. OMG Systems Modeling Language (OMG SysML) V1.1 [Online]. Object Management Group. Available: http://www.omg.org/spec/SysML/1.1/PDF [Accessed 16 May 2009].

A TRIZ Based Methodology for the Analysis of the Coupling Problems in Complex Engineering Design

G. Fei[1], J. Gao[1], X.Q. Tang[2]
[1]School of Engineering, University of Greenwich, Chatham Maritime, Kent, ME4 4TB, UK
[2]School of Mechanical Engineering and Automation,
Beijing University of Aeronautics and Astronautics, Beijing, China
J.Gao@gre.ac.uk

Abstract

Conceptual design is a critical and innovative stage in engineering product and system design. In the conceptual design process, it would be ideal if all functional requirements are maintained independently according to the law of Axiomatic Design theory. However, in practice, especially in complex engineering product and system design, more often the requirements are not independent (or coupled), and this makes conceptual design more difficult. In this paper, a coupling analysis methodology, framework and related techniques are proposed which integrate axiomatic design with the theory of inventive problem solving (TRIZ), in order to identify and analyse the coupling problems existing in conceptual design. An illustrative example is also presented.

Keywords:

New product design, Coupling analysis, Axiomatic design, TRIZ

1 INTRODUCTION

Conceptual design, seen as a critical part of new product and system development, is getting more attention both in academia and industry. Many techniques have been proposed in the past decades in order to improve the effectiveness and efficiency of conceptual design. Some of them, such as Quality Function Deployment, Axiomatic Design [1], and the Theory of Inventive Problem Solving (TRIZ) [2], are proven successful in conceptual design of engineering products and systems and are widely used in industrial applications. Axiomatic design provides an effective approach to developing products and systems throughout the whole design domains, including the customer domain, the functional domain, the physical domain and process domain. The zigzagging developing process and two axioms of axiomatic design theory developed by Suh [1] are widely adopted, especially in mapping functional requirements to design parameters at the conceptual design stage. During the zigzagging process, function requirements and design parameters are acquired with corresponding design matrices. By populating design matrices, uncoupled, decoupled and coupled solutions can be identified and further measures can be carried out to eliminate couplings. However, this is not viable in some complex engineering products and systems in the real world [3,4,5]. Firstly, current techniques of coupling analysis are implemented on a qualitative basis. The strengths of couplings existing in the solution can not be obtained. For example, when there are many couplings and not all of them can be solved all together, the critical couplings need to be identified, prioritised and solved in order to improve the effectiveness and efficiency of design. Therefore, it is important to find a methodology that can analyse and quantify the strengths of couplings. Secondly, the original theory of axiomatic design is inefficient when the scale of design matrix gets very large. Generally, decoupling of design is conducted in two steps, i.e., (i) the design matrix is populated so that couplings existing in the design are identified; and (ii) the design matrix is rearranged to adjust the sequence of functions and design parameters in order to make the design decoupled. However, when the number of functional requirements increases, the number of combinations will grow in a geometric progression and the rearrangement of design matrix will be extremely time consuming [1], and this is difficult to implement in industry. Thirdly, resources, in terms of development costs, lead-time, staffing and so on are always precious and need to be allocated properly in most projects. The scale and complexity of some large engineering projects are enormous and solutions of these couplings are not easy to be obtained. Therefore, it is unacceptable to spend too much resource in resolving the less critical coupling issues (some are even harmless). Instead, the critical couplings should be identified and resolved with intensive efforts.

In summary, a more practicable and efficient coupling analysis approach is needed to analyse the couplings existing in design solutions, which is able to identify couplings quickly and enables engineers to make more efforts on solving critical couplings that are most harmful to the implementation of required functions. In addition, the progress of the project may be speeded up and unnecessary costs may be reduced by leaving the less critical or even harmless couplings unsolved.

2 LITERATURE REVIEW

2.1 Axiomatic Design

The theory of axiomatic design is proposed by Suh [1], which is dedicated to constructing a design framework with a scientific basis and improving design activities with a logical and analytic thinking process. Basically, there are three essential parts of the axiomatic design that are widely used in academic research and industrial applications, namely the zigzagging design process, design axioms and the design matrix. The Axiomatic design theory divides the design world into four domains, i.e., the customer domain (CAs), the functional domain (FRs), the physical domain (DPs) and the process domain (PVs). The design is gradually realised by mapping from one domain to another. Typically, the mapping process between functional domain and physical domain is studied more often in literature than others because the conceptual design is mostly undertaken at this stage. As depicted in Figure 1, the mapping system works in a top-down way. Each design parameter (DP) in the physical domain corresponds to each functional requirement in the functional domain at the same level. Then design parameters in this level derive functional requirements in the next level until it reaches the leaf level so that functions and solutions are decomposed and obtained during this process.

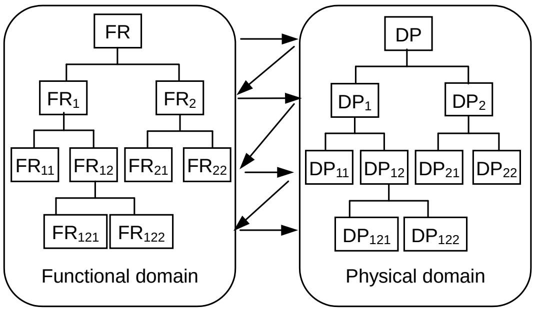


Figure 1: Zigzagging mapping process between functional domain and physical domain [1]

There are two axioms recognised in design, namely independent axiom and information axiom, companied by related theorems and corollaries. Design axioms are the elementary part of axiomatic design and deemed as the basis of good design, which are used to guide the design process and evaluate alternative solutions. The independent axiom indicates that the function requirement should always be maintained independently so that any change of the corresponding DP of one FR will not affect functionalities of other DPs. As the basis of the axiomatic design theory, the independent axiom takes effect throughout the design process. The information axiom indicates that the best design solution should contain minimum information content. More information means being more complicated and more possible that the design parameter can't satisfy the functional requirement.

Design matrix is a technique used to analyse the coupling relationships between a group of FRs and their corresponding DPs. Normally the matrix is populated in a binary way so that all the coupling relationships are recognised qualitatively. According to the independent axiom, only uncoupled and decoupled designs are acceptable. However in the design of some complex engineering products and systems, it is impossible to keep all FRs independent of DPs. Quantitative analysis of coupled elements should be carried out. A practical approach is needed to clarify the coupling relationships within these designs so that the direction of improvement can be pointed out.

2.2 The Substance-Field Model of TRIZ

TRIZ is the Russian acronym of Theory of Inventive Problem Solving [2]. Since TRIZ was proposed, it has been widely used in industrial applications to solve technical problems due to the fact that TRIZ is a result from the analysis of thousands of patents. Recently, many researchers and practitioners are trying to apply TRIZ in other non-technical areas, such as management, education, environment and politics. Although there are a considerable set of techniques in the theory of TRIZ, such as contradiction matrix, inventive principles, knowledge/effects and ARIZ, the Substance-Field (shortly Su-Field) analysis model is picked up in this project in order to clarify the coupling relationships during the zigzagging design process of Axiomatic design. The Su-Field analysis model is based on the minimal technological system which is also known as the triad 'object-tool-energy'. The triad system is composed of a tool, an object and the energy and describes that the tool performs action on the object by the force coming from the energy. Through the analysis of the triad system, interactions between elements within this system can be clarified. Along with the triad system, four kinds of actions are also identified which include unspecified action, specified action, inadequate action and harmful action. For example, the Su-Field analysis model of driving nail into the wall is depicted in Figure 2. In this system, mechanical force is performed on the hammer by the user, and then the hammer performs mechanical force on the nail.

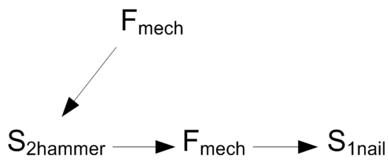


Figure 2: Su-Field model example

In this project, direct or indirect interactions between DPs in the Axiomatic design methodology may be identified using the Su-Field analysis method. Fields existing in interactions can be clarified as well so that effects caused by fields can be estimated by specific expertise. This is important to identify the couplings between DPs and FRs.

2.3 Integration of Axiomatic Design and TRIZ

Many attempts at integrating TRIZ and Axiomatic design together have been made by researchers, in order to improve the product development process. Comparison between Axiomatic design and TRIZ is carried out to identify advantages and disadvantages of the two theories. The possibility of complementary integration of axiomatic design and TRIZ is also discussed [6,7,8]. It is found that, on one hand the Axiomatic design is powerful in functional analysis and provides a logical thinking approach to devising conceptual design in a zigzagging and hierarchical structure. On the other hand, although it is effective to identify functional conflicts underlying the solutions, there is a lack of specific tools in the Axiomatic design theory for problem solving [9]. Based on a wide range of analysis of a large number of patents, TRIZ becomes a sophisticated methodology for physical and technological problem solving. However, it is relatively less powerful in complex system analysis [6, 10]. With the advantages of TRIZ, it is possible to improve the ability of problem identification and solving within the Axiomatic design theory.

In the light of the above discussions, many methodologies have been proposed to enhance the capability of product design by making Axiomatic design and TRIZ work together [5,8-13]. Particularly, some methodologies are devised, from different perspectives, for coupling analysis recently. Su and his colleagues [4] developed a methodology to deal with coupling analysis of engineering system design in a quantitative way. A comparative approach and a scale algorithm are proposed in order to transfer the binary design matrix into a quantitative one on an analytical basis. Zhang et al [5] proposed a conceptual framework by integrating TRIZ with axiomatic design. Some tools of TRIZ, such as contradiction analysis, separation principles, inventive principles and effects, are used to solve constraints and coupling problems. Shin and Park [13] classified the coupled designs into six patterns. Tools of TRIZ, such as standard solutions, scientific and technical effects, contradiction matrix, separation principles and ARIZ, are used in each pattern respectively or combined, to solve different coupling problems. Kang [12] proposed an uncoupling methodology using contradiction matrix and inventive principles. Within this methodology, coupling problems are formulised as contradictions and FRs are converted into standard characteristics, and then inventive principles are applied to solve all the contradictions.

By reviewing the above methodologies regarding the integration of Axiomatic design and TRIZ, it is found that there still exits a weakness in using these methodologies in conceptual design. TRIZ is good at solving technical and physical problems, but in conceptual design, detail design parameters are still vague and it is difficult, and also time-consuming, to solve problems using the principles or standard solutions in TRIZ. The aim of this project is to identify the coupling relationships within solutions and find critical paths for designers to focus on. As an ongoing project, although not all the coupling problems will be solved in the proposed methodology directly, it provides an efficient way for designers to find which path is most valuable to take for improvement.

3 THE PROPOSED COUPLING ANALYSIS METHODOLOGY

Functional design is to find an object or a group of objects that can realise the function requirements by some properties of them or by interactions between them. In other words, the function is the outcome of the operation of the triad system, in terms of TRIZ. Design parameter is one kind of properties of these objects that can be used to drive the realisation of required functions. Any unexpected actions will affect the realisation of functions. In this project, expected interactions within and between design elements that are used to realise functions are not considered. Instead, unexpected interactions are focused on because they are most possible to cause unexpected couplings. Different from the term "contradiction", unexpected interactions are not contradictions or conflicts from a technical point of view. They are just functional interactions, but out of the expectation of designers.

The approach to analysing couplings in the zigzagging design process is depicted in Figure 3. As the product design is organised in a hierarchical structure by the zigzagging process, design parameters (DPs) in lower levels should be consistent with their parent ones (parent-DPs) in upper levels. In other words, characteristics of design parameters in lower levels will reflect characteristics of those in upper levels. Given the coupling analysis is carried out in the second level of the zigzagging process, design parameters DP_{11} and DP_{12} are identified as coupled in this solution. On account of the lack of design details at this stage, although the qualitative results of impact of this coupling can be roughly estimated by the inputs and outputs of DP_{12} and DP_{11}, the more accurate and quantitative strength of coupling can not be obtained yet. Provided that the third level is the leaf level of this design, the corresponding child design parameters of DP_{11} are DP_{111}, DP_{112} and DP_{113}, and likewise, the corresponding child design parameters of DP_{12} are DP_{121} and DP_{122}. At the leaf level, behaviours of these child design parameters are analysed by the Su-Field analysis model, so that couplings between design parameters derived from the same parent parameters are identified and quantified. At the same time, couplings between child parameters of different parent parameters are also identified. Pointing to the second level, by analysis of the third level of design parameters, not only couplings within DP_{11} and DP_{12} can be calculated by specific algorithms but also coupling between DP_{11} and DP_{12} , which is caused by F_{12}(o'), can be determined by analysing behaviours between their child parameters (i.e. F_{121}(o)' and F_{122}(o)').

Due to the fact that this project primarily focuses on the analysis of coupling relationships between design parameters, the Su-Filed method is partially used. Conventionally, Su-Field method is used to analyse problems and guide designers to solving problems with standard solutions [2]. In this project, standard solutions are not involved, because no efforts will be made to solve coupling problems at this stage. In other words, the triad analytical model is the only part that is used to clarify interactions within solutions. Discussion of using standard solutions or laws of system evolution to suggest or predict the measure of improvement is out of the scope of this project.

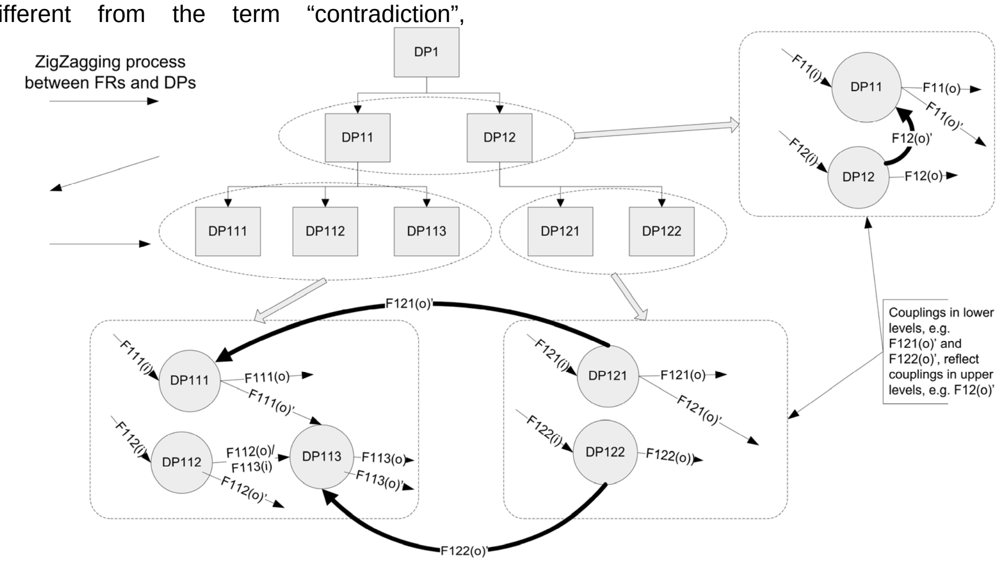


Figure 3: Analysis in the Zigzagging design process

4 COUPLING ANALYSIS TECHNIQUE WITH SU-FIELD METHOD

In order to formulate the coupling analysis process, the framework of coupling analysis methodology is developed (see Figure 4), which is mainly composed of 8 steps. In this section, every step of this framework will be described and related techniques used in each step will be clarified.

Step1: Complete the zigzagging design process. The zigzagging design process is conducted by designers at the beginning of product design. Hierarchical design structures of functional requirements (FRs) and corresponding design parameters (DPs) are constructed with current design capability of the team. A qualitative design matrix is populated and rearrangement of the matrix is conducted so that uncoupled and decoupled functions are identified [14]. Meanwhile, coupled blocks existing in the binary design matrix are identified as well, which are looked into in this project. Unlike the conventional axiomatic design approach that does not decompose coupled blocks, in the proposed methodology, each coupled block in the design matrix is decomposed further until it reaches the leaf level and interactions between constituent elements are analysed in step 2 by the Su-Field analysis method.

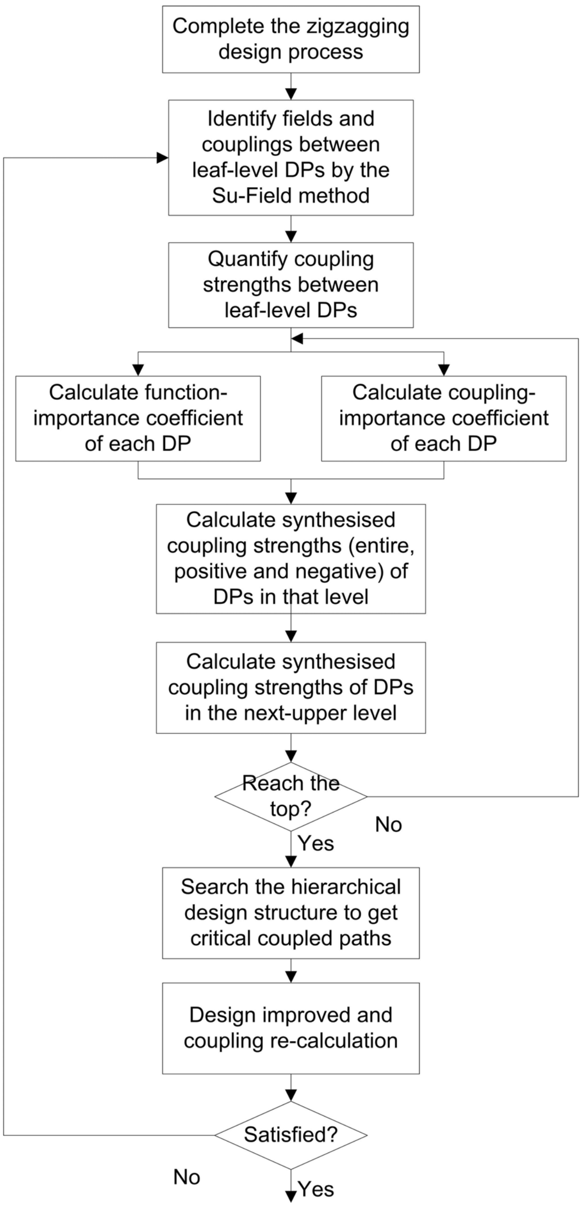


Figure 4: The framework of coupling analysis

Step2: Analyse couplings between leaf-level DPs by the Su-Field method. The coupling analysis method in this project is built upon the Su-Field analysis method which is used to clarify interactions between design elements and their effects caused by these interactions. For example, as depicted in Figure 5, there are three DPs and their interactions are expressed in the way of Su-Field analysis. In this coupling analysis model, Fields are denoted by F(i), F(o) and F(o)', and Substances are denoted by DP.

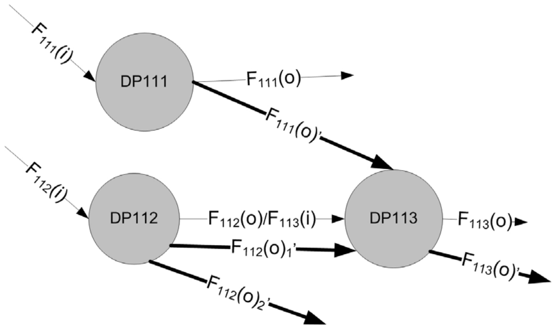


Figure 5: Su-Field analysis of couplings among Design Parameters

F(i) denotes the expected input field of a DP, which is designated when the DP is designed. F(i) could be fields coming from out of the system, like actions from users or environments, or fields coming from other DPs in the system. F(o) is the expected output field of a DP. Similarly to the F(i), F(o) is also designated when the DP is designed. The F(o) is what the system wants in order to realise the function corresponding to the DP. F(i) and F(o) are necessary for the realisation of functions, so in this paper the couplings caused by F(i) and F(o) are not considered. Another output field is F(o)' which is not expected by the initial design of the system. In other words, F(o)' is the factor that may be out of control and cause unexpected couplings between DPs. So the analysis of F(o)' will clarify what the coupling of DPs is, how the coupling happens.

Another important factor in this model is the DP. Strictly according to the theory of Axiomatic design, DP means a feature that can satisfy the realisation of functional requirement. The carrier of desired feature may be an object or a particular part of an object. For simplicity, here, 'DP' denotes an object or a part of an object that has these design parameters so that the expression can be consistent with the theory of Su-Field analysis as well. In terms of design parameters, their expected states are controlled by F(i)s and their carriers. However, with influence made by F(o)'s from other design parameters, their states may vary. Thus, by comparing the state influenced by F(o)'s with the initial state expected by design, changes of these states of DPs are looked into. The effects of functional performance caused by changes of DP's states can be quantified by a scale system so that strengths of couplings can be obtained.

Step3: Quantify coupling strengths between leaf-level DPs. Due to the fact that couplings are caused by unexpected fields, i.e., F(o)', acting on DPs, the effort of calculating coupling strength is focused on the influence that F(o)'s make on DPs. To achieve that, a scale system is developed. The strength of coupling is scaled by engineering experts according to the effect that one DP performs on another DP in every level of the zigzagging design process. The relationship between coupling strengths and effects can be learnt from Table 1. Taking the system in Figure 5 as an example, if F_{113}(o)' performs a negative effect on DP_{113}, which significantly reduces it's performance, then the scaled coupling strength will be marked as -5 on DP_{113}; if the $F_{112}(o)_1$' performs a positive effect on DP_{113}, which slightly improves it's performance, then the scaled coupling strength will be marked as 1 on DP_{113}, as depicted in Figure 6. Along with the progress of zigzagging design, the scale system expresses the coupling strength in a more accurate way, because there are more details emerged from top level to lower level design until the leaf level. In turn, more accurate estimation of coupling strength in lower levels can improve estimation of coupling strength in upper levels with the help of an estimating algorithm.

Coupling Strengths	Descriptions of Coupling Strengths
9	Necessity of function
7	Extreme performance improvement
5	Significant performance improvement
3	Moderate performance improvement
1	Slight performance promotion
0	No effect
-1	Slight performance reduction
-3	Moderate performance reduction
-5	Significant performance reduction
-7	Extreme performance reduction
-9	Function damaged

Table 1: The scale system of coupling analysis

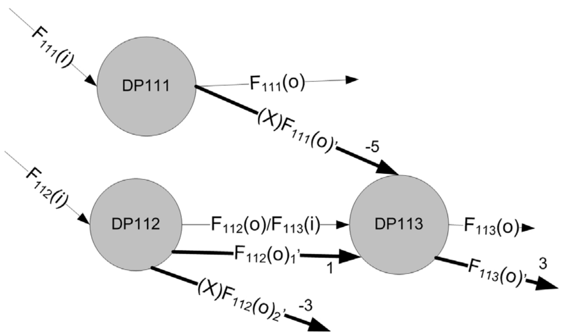


Figure 6: Example of scaled coupling strength

Step4: Calculate relative importance of each DP. Before calculating the coupling strength of a DP, the importance of each DP needs to be clarified, due to the fact that DPs have different importance compared to each other. DPs with different importance will be considered differently when their coupling strengths are calculated. In this project, there are two kinds of importance need to be analysed, which are the functional importance and the coupling importance.

Step 4.1: Calculate the functional importance. Among the child design parameters of the same parent parameter, one child parameter is expected to realise one child function of the corresponding parent function. Obviously, these child functions play different roles in realising the parent function therefore there is different relative importance existing. To obtain the relative importance of each child function, the Analytical Hierarchical Process method (AHP) [15] is used to deal with a pair-wise comparison between these child functions of a parent function. As a result, each child parameter will get a relative importance coefficient which will be used in calculating its coupling strength. For DP_l, its relative importance coefficient is denoted as ε_l, where $\varepsilon_l \in (0,1)$ and DP_l means a certain DP in the hierarchical structure, e.g. DP_{113} in Figure 5.

Step 4.2: Calculate the coupling importance. When a DP performs actions on another DP, it means this DP has the ability to influence others. Given that there is a DP_1 performing actions on DP_2, in other words there is a coupling between them, the outcome of DP_2 will be influenced by DP_1. Furthermore, the outcome of DP_2 will act on other DPs that are coupled with DP_2. Thus, it is important to consider the ability that how one DP can influence others before calculating its coupling strength. The coupling importance coefficient of DP_l is denoted as λ_l which can be calculated as follows:

Provided that DP_l has *K* F(o)'s act on *H* DPs, each single coupling strength resulting from F_k(o)' acting on DP can be denoted as f_k and the functional importance of each DP that is acted on by F(o)'s is denoted as ε_h, where $k \in K$ and $h \in H$. Then, the original coupling importance can be calculated as:

$$\hat{\lambda}_l = \sum_{(h-1)|k}^{H} \sum_{k=1}^{K} |f_k| \bullet \varepsilon_h \quad (1)$$

In order to be consistent with functional importance, the importance coefficient should be a number between 0 and 1. Thus, the original coupling importance needs to be normalised. The normalised coupling importance coefficient can be calculated as:

$$\lambda_l = \frac{\hat{\lambda}_l}{\sum_{l=1}^{L'} \hat{\lambda}_l} \quad (2)$$

where *L'* denotes the number of child parameters of DP_l's parent parameter.

Step5: Calculate synthesised coupling strengths of DPs. The coupling of DP_l can be expressed by $C_l(ct)_{cn}^{cp}$, where ***cp*** means the aggregate coupling strength caused by positive effects performed on DP_l, ***cn*** means the aggregate coupling strength caused by negative effects performed on DP_l, and ***ct*** means the aggregate coupling strength caused by all effects performed on DP_l. For example, if there are *n* fields act on DP_l, *p* of them make positive effects on DP_l and *q* of them make negative effects on DP_l. Then cp_l and cn_l can be calculated as follows:

$$cp_l = \varepsilon_l \bullet \lambda_l \bullet \sqrt{\sum_{i-1}^{p} f_i^2} \quad (4)$$

$$cn_l = \varepsilon_l \bullet \lambda_l \bullet \sqrt{\sum_{j=1}^{q} f_j^2} \quad (5)$$

where $i \in \{0,...,p\}$, $j \in \{0,...,q\}$, $p+q=n$ and f means the coupling strength caused by a field; The aggregate coupling strength can be calculated by this equation:

$$ct_l = \sqrt{cp_l^2 + cn_l^2} \quad (6)$$

Step6: Calculate synthesised coupling strengths of DPs in the next-upper level. For the parent design parameter, its coupling strength can be calculated easily by integrated coupling strengths of child parameters together. For example, if DP_p has R child parameters, then the coupling strength can be calculated as follows:

$$cp_p = \sqrt{\sum_{r=1}^{R} cp_r^2} \quad (7)$$

$$cn_p = \sqrt{\sum_{r=1}^{R} cn_r^2} \quad (8)$$

$$ct_p = \sqrt{cp_p^2 + cn_p^2} \quad (9)$$

where cp_p denotes the aggregate positive coupling strength of the parent DP, cn_p denotes the aggregate negative coupling strength of the parent DP, cp_r denotes the aggregate positive coupling strength of a child-DP, and cn_r denotes the aggregate negative coupling strength of a parent-DP. The coupling strength of every DP in each level is calculated until it reaches the top level. Before calculating the coupling strength of each upper level, relative importance coefficient needs to be calculated first.

Step7: Search for the hierarchical design structure to get critical coupled paths. After obtaining all the coupling strength of every DP in each level, a searching algorithm is used to identify critical coupling paths in this hierarchical design structure. Designers can get the most coupled path by searching coupling strengths *t* from the top level to the leaf-level in order to get the most promising route to improve the design. Designers can also get the most negative coupled path by searching negative coupling strengths *cn* in the structure in order to get the most valuable way to eliminate critical problems existing in the design. Additionally, designers can get the most positive coupled path by search for the positive coupling strength *cp* of every DP so that they can decide whether some parts of the design can be integrated together.

Step8: Design improved and coupling re-calculation. By recognising some most valuable paths for improving the design, improvements need to be implemented and the design is refined. If the design is still not satisfactory, recalculation of the coupling strength of the design is carried out and further improving work needs to be done.

5 AN ILLUSTRATIVE EXAMPLE

In this section, an example is demonstrated to show how the methodology works to identify and quantify the coupling relationship between FRs and DPs in an engineering system. The engineering system chosen in this paper is the reactor cavity cooling system (RCCS) of General Atomics' Gas Turbine-Modular Helium Reactor (GT-MHR) nuclear reactor which is described in the GT-MHR conceptual design description report [16,17] and is further studied by Jeff Thielman et al [3,18] in order to evaluate and optimise the system with Axiomatic design theory. The RCCS is one of the cooling systems of the GT-MHR and works in a passively natural circulation cooling condition to remove decay heat when the reactor is shut down (see [10] and [12] for details). Although there are seven sub-FRs and seven sub-DPs of the DP3.2.2 in Jeff Thielman's research, for the purpose of demonstrating the proposed methodology in the simplest way, only three FRs and their corresponding DPs are selected in this paper, which can be found in Table 2.

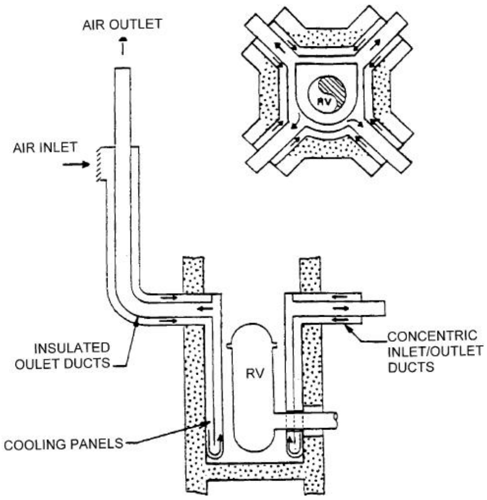


Figure 7: The reactor cavity cooling system [18]

Sub-FRs of FR3.2.2	Sub-DPs of DP3.2.2
Air exit temperature	Riser Width
Air velocity in riser	Riser Height
Maximum riser wall temperature	Outlet Area

Table 2: Selected FRs and DPs of RCCS

There are three functional requirements selected in this demonstration, namely air exit temperature, air velocity in riser, and maximum riser wall temperature. In the conceptual design of the RCCS, the air exit temperature is supposed to be maintained as low as possible. The value of air velocity in riser needs to be kept as high as possible so that there will be more heat taken from the reactor. Obviously, the maximum riser wall temperature is designed to be as low as possible due to the fact that high temperature is negative to the safety of the reactor.

The operation model of the RCCS system is simply built up using the Su-Field analysis method, as depicted in Figure 8. Obviously, it can be learnt that Reactor is the source of heat and it delivers heat to risers by radiation. The riser, therefore, is heated and delivers forward the heat to the circulating air in the riser. By the nature of air, the heated air drives air in the riser to rise up and go outside of the riser. Finally, the exit air is led by the outlet duct and gets into the atmosphere.

Beside these expected actions, there are also some actions that are not expected by the original design. For example, with increase of the width of riser, the air inside the riser is heated more effectively so that the velocity of air in riser increases. Meanwhile, the temperature of the riser wall decreases because there is more heat taken from the reactor. Another fact is when the height of riser increases the temperature of exit air and the maximum temperature of the riser wall increase because when the height of riser increases the damp of air circulation increases as well and the performance of releasing heat decreases. The outlet area also affects functions of air exit temperature and air velocity in riser by control of the exit of air. Thus, the coupling diagram can be obtained based on the analysis of Su-Field method and engineering expertise, which is shown in Figure 9. By analysing the effects caused by unexpected actions, relative coupling values are obtained according to the scale system of coupling analysis.

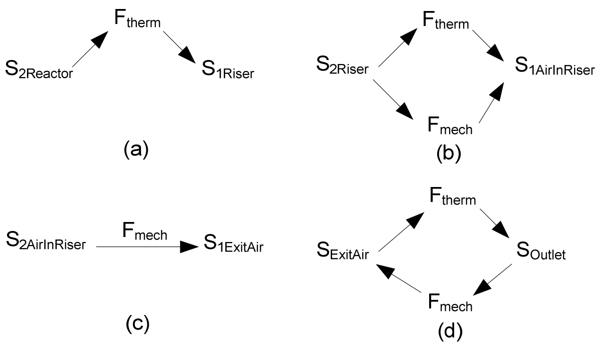


Figure 8: Su-Field analysis of heat removal

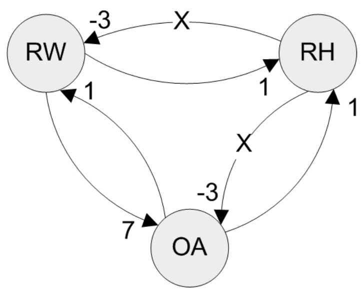


Figure 9: Interactions and couplings between objects

After obtaining the coupling relationship between design elements, step 4 of the coupling analysis framework needs to be carried out to calculate the relative functional importance and relative coupling importance of each element. By the algorithm of AHP (Analytical Hierarchical Process), the relative functional importance can be calculated as in Table 3.

	Air exit temperature	Air velocity in riser	Maximum riser wall temperature	Relative Importance
Air exit temperature	1	2	1/2	0.297
Air velocity in riser	1/2	1	1/3	0.164
Maximum riser wall temperature	2	3	1	0.539

Table 3: Relative functional importance

According to the results of coupling analysis in Figure 9 and the relative functional importance in Table 3, the relative coupling of each design element can be obtained by equation 1 and equation 2, which is shown in Table 4.

	RW	RH	OA
Relative coupling importance	0.57	0.067	0.363

Table 4: Relative coupling importance

Furthermore, the coupling value of each design element can be calculated by equation 4, 5, 6. The results are shown in Table 5.

	Positive coupling (cp)	Negative Coupling (cn)	Total coupling (ct)
RW	0.169	-0.508	0.535
RH	0.011	-0.011	0.016
OA	1.37	-0.587	1.49

Table 5: Coupling strengths of design elements

Finally, the coupling strength of the parent element, DP3.2.2, can be calculated by equation 7, 8, 9. The result of coupling strength of DP3.2.2 is displayed in Table 6. It needs to be noticed that the coupling strengths of DP3.2.2 below are not the actual values because there are only three pairs of FRs and DPs selected to demonstrate in this example.

	Positive coupling (cp)	Negative Coupling (cn)	Total coupling (ct)
DP3.2.2	1.38	-0.776	1.583

Table 6: Coupling strength of DP3.2.2

By calculation of coupling strengths of three design elements, the coupling problem can be learnt from the result intuitively. From Table 5, the design element OA is supposed to the critical element that gets coupled in the system with others because both the strongest negative coupling and the strongest total coupling occurred on OA. Thus, some proper improving efforts should be assigned to the design of OA in order to effectively reduce the couplings of the solution. If the full decomposition of the design structure and the coupling analysis of all design elements are completed, there would be a hierarchical structure of coupling analysis results where a comparative algorithm can be applied to search the strongest couplings in each level in a top-down way. As a result, critical paths for system improvement are identified to facilitate the effectiveness and efficiency of product design.

6 CONCLUSIONS

The theory of Axiomatic design is widely used in new product and system design, especially at the conceptual design stage. According to the Independence Axiom, it is critical to maintain the independence of functions that minimises the disturbance to realisations of other functions when anyone of design parameters changes. However, in the real world, it is almost impossible to maintain the complete independence of all functions at an acceptable cost in some complex engineering systems. In this project, a methodology of coupling analysis is proposed by integrating TRIZ with Axiomatic Design. Su-Field method, an important part of TRIZ, is used to identify and analyse the couplings existing in design solutions. With the assistance of this methodology, coupling relationships within the designs are clarified and quantified. It is much easier for designers to find out clues to improve the system. Furthermore, if the number of design parameters is large, it is impossible for designers to carry out a rearrangement of the design matrix. Therefore this method can help to find critical coupled elements that affect the performance of the system. Also, it can help to improve the effectiveness and efficiency of engineering design because critical coupled paths can be found by searching in the hierarchical structure based on the coupling analysis results. The design team can make more efforts to improve the critical aspects of the system (and less efforts on less important or harmless couplings) and resources can thus be allocated more properly.

7 DISCUSSION AND FURTHER WORK

Although the proposed methodology provides a new way to analyse coupling issues in conceptual design, some uncertainties and shortcomings also appear, which are worthy of further discussion and consideration. From the perspective of TRIZ, substance in the Su-Field method indicates “thing” or “entity” which normally is a physical object. In this project, the carrier of design parameters is considered as an object or a part of an object which possesses the feature that can realise the corresponding function. Therefore, design parameter is used to represent that object or that part of the object or carrier. However, the carrier may have more than one feature to realise different functions. Thus, further analysis needs to be carried out to clarify which action is performed on a certain feature and what is the effect. This analysis is done by individual designers in this paper. A further research of mapping between physical Su-Field analysis and abstract coupling analysis is interesting to be looked into. Another issue is that the scaling system of coupling strength is used to quantify the coupling based on the expertise of individual engineers, which may make the estimation of couplings inconsistent if there are engineers in the team with different levels of experience. The scientific and technical effects of TRIZ are possible to be helpful to estimate the coupling strength by analysing the interactions. In this paper, the coupling strength of design element is the value that denotes the effects caused by other design elements acting on the current design element. But the effect, which is caused by the current design element acting on other design elements, has not been considered. Further research needs to be done in order to clarify the strengths of effects that the current design element acts on other design elements. The illustrative example in this paper is based on a complex engineering system. However, due to progress of the current research, the system has not been decomposed in details so that coupling analysis is not based on a rigorous engineering analysis and coupling analysis in upper-levels is not demonstrated. Thus, further research on the reactor cavity cooling system needs to be carried on. A real industrial case study is planned in the next stage of this project.

REFERENCES

[1] Suh, N.P., 2001, Axiomatic Design: Advances and Applications, Oxford University Press, New York.

[2] Fey, V. and Rivin, E., 2005, Innovation on Demand: New Product Development Using TRIZ, Cambridge University Press, New York.

[3] Thielman, J. and Ge, P., 2006, Applying axiomatic design theory to the evaluation and optimization of large-scale engineering systems, Journal of Engineering Design, 17(1): 1-16

[4] Su, J.C., Chen, S.J. and Lin, L., 2003, A structured approach to measuring functional dependency and sequencing of coupled tasks in engineering design, Computers and Industrial Engineering, 45: 195–214.

[5] Zhang, R., Tan, R. and Li, X., 2005, An innovative conceptual design model using axiomatic design and TRIZ, The 15th CIRP International Design Seminar, Shanghai, China, 22-26 May: 281-286.

[6] Lee, K.W., 2005, Mutual Compensation of TRIZ and Axiomatic Design, the proceedings of the European TRIZ Association Conference, TRIZ Futures 2005, Graz, Austria, November

[7] Mann, D, 2002, Axiomatic Design and TRIZ: Compatibilities and Contradictions, Proceedings of ICAD2002, Second International Conference on Axiomatic Design, Cambridge, MA, June 10-11

[8] Kim, Y.-S., 2000, Reviewing TRIZ from the perspective of Axiomatic Design, Journal of Engineering Design, 11(1): 79-94

[9] Hua, Z., Yang, J., Coulibaly, S. and Zhang, B., 2006, Integration TRIZ with problem-solving tools: a literature review from 1995 to 2006, International Journal of Business Innovation and Research, 1(1/2):111 - 128

[10] Yang, K., Zhang, H., 2000, Enhancing Robust Design with the Aid of TRIZ and Axiomatic Design, Proceedings of ICAD2000, International Conference on Axiomatic Design, Cambridge, MA, June 21-23

[11] Shirwaiker, R.A., and Okudan, G.E., 2008, Triz and axiomatic design: a review of case-studies and a proposed synergistic use, Journal of Intelligent Manufacturing, 19(1):33-47

[12] Kang, Y.J., 2004, The Method for Uncoupling Design by Contradiction Matrix of TRIZ, and Case Study, Proceedings of ICAD2004, The Third International Conference on Axiomatic Design, Seoul, June 21-24

[13] Shin, G.-S., Park G.-J., 2006, Decoupling process of a coupled design using the TRIZ, Proceedings of ICAD2006, the 4th International Conference on Axiomatic Design, Firenze, June 13-16

[14] Gabala, D.A. and Eppinger, S.D., 1991, Methods for Analyzing Design Procedures, ASME Conference on Design Theory and Methodology, Miami, FL, September: 227-233.

[15] Saaty, T.L., 1999, Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World, RWS Publications, Pittsburgh, Pennsylvania.

[16] General Atomics, 2002, Gas Turbine-Modular Helium Reactor (GT-MHR) Conceptual Design Description Report - Part 1, U.S. Nuclear Regulatory Commission, http://www.nrc.gov/reactors/new-licensing/new-licensingfiles/gtmhr-preapp1.pdf.

[17] LaBar MP., 2002, The gas-turbine modular helium reactor: a promising option for near-term deployment. International Congress on Advanced Nuclear Power Plants, ICAPP 2002, Hollywood, Florida, USA, 9-13 June: GA-A23952

[18] Thielman, J., Ge, P., Wu, Q. and Parme, L., 2005, Evaluation and optimization of General Atomics' GT-MHR reactor cavity cooling system using an axiomatic design approach, Nuclear Engineering and Design, 235:1389–1402