Design Automation Systems – Supporting Documentation and Knowledge Management

Jie Nan
Qian Li

This exam work has been carried out at the School of Engineering in Jönköping in the subject area Product Development and Materials Engineering. The work is a part of the Master of Science programme. The authors take full responsibility for opinions, conclusions and findings presented.

Examiner: Joel Johansson
Supervisor: Fredrik Elgh
Scope: 30 credits
Date: 2012-09-21
Abstract

During the practical use of Design Automation (DA) System in a company, the lack of assistance from either documentation work about the whole system or management of knowledge could bring out some obstacles when engineers reuse existing knowledge and information. The purpose of this project is to explore an approach of documentation and knowledge management in DA System. The study is mainly based on the actual case of seat heater DA system developed by JTH.

Based on preset functional requirement for the potential solution, several principles and methods of documentation and knowledge management are introduced such as MOKA, CommonKADS, SysML and PVM. A number of useful applications such as DRed (Design Rationale Editor), PC PACK, Sementic MediaWiki and Product Model Manager became candidates solutions for this project. The selection of final approach was Sementic MediaWiki, and this is based on the comparison of the result from evaluation of functionality of each application.

Due to specificity of documentation on the DA system, the “process based” approach had been used for structuring system included knowledge instead of using a systematical method like either MOKA or CommonKADS completely. Setting up interconnection between different knowledge objects was one of the most important tasks in this project because it enables capturing and retrieving of knowledge.

Sementic MediaWiki, a powerful text representative and web-based tool has been used as a platform of representing the whole knowledge and information. With its implementation, the performance of Sementic MediaWiki had been tested according to the preset functional requirement. After a slight refine process to the solution, the satisfactory result had been achieved, and also proved the applicability of Sementic Wiki in such kind of project.

Keywords

Documentation, Knowledge Management, Semantic MediaWiki, Design Automation system.
# Contents

1 Introduction ................................................................. 4
   1.1 BACKGROUND ......................................................... 4
   1.1.1 Background of R&D ........................................ 4
   1.1.2 Theoretical principles ....................................... 4
   1.1.3 Introduction of ProcedoStudio ............................ 6
   1.2 PROBLEM DEFINITION .......................................... 9
      1.2.1 Problems, need and company’s perspective .... 9
      1.2.2 Academic perspective .................................. 10
   1.3 PURPOSE AND RESEARCH QUESTIONS ................. 12
   1.4 DELIMITATIONS .................................................... 12
   1.5 OUTLINE ............................................................. 13

2 Frame of Reference ................................................... 14
   2.1 PRODUCT DEVELOPMENT ................................. 14
   2.2 ENGINEERING DESIGN ....................................... 14
   2.3 DESIGN AUTOMATION ......................................... 16
   2.4 KNOWLEDGE-BASED ENGINEERING (KBE) .... 17
   2.5 DESIGN RATIONAL AND DOCUMENTATION .... 18
   2.6 SUMMARY .......................................................... 19

3 Functional Requirement ............................................. 20
   3.1 DEFINITION OF REQUIREMENT .................... 20
   3.2 DEFINITION OF REQUIREMENT .................... 20

4 General Principles and Applications .......................... 24
   4.1 INTRODUCTION OF PRINCIPLES .................... 24
      4.1.1 SysML ......................................................... 24
      4.1.2 MOKA ......................................................... 27
      4.1.3 CommonKADS ........................................... 29
      4.1.4 PVM ......................................................... 30
   4.2 INTRODUCTION OF APPLICATIONS ................ 32
      4.2.1 Design rational editor ................................ 33
      4.2.2 PC Pack .................................................... 34
      4.2.3 Semantic MediaWiki .................................. 35
      4.2.4 Product model manager ............................ 36
   4.3 SUMMARY .......................................................... 38
   4.4 IDEAS OF ALTERNATIVE APPROACH ............ 38
   4.5 EVALUATION AND SELECTION OF APPROACH .... 38

5 Development and Implementation .............................. 39
   5.1 COLLECTING KNOWLEDGE AND INFORMATION .... 39
   5.2 STRUCTURING KNOWLEDGE .............................. 40
      5.2.1 Design sequence electric ......................... 42
      5.2.2 Design sequence geometry ....................... 46
   5.3 IMPLEMENTATION .............................................. 56
   5.4 TEST AND EVALUATION .................................. 62
   5.5 APPLICABILITY AND GENERALIZATION ........... 66

6 Discussion and Conclusions ................................... 67
   6.1 DISCUSSION OF SOLUTION ............................. 67
   6.2 DISCUSSION OF METHOD ................................. 67
   6.3 CONCLUSIONS AND FUTURE WORK ............... 67
7 Reference .................................................................................................................. 69


1 Introduction

The objective with this project is to explore approaches for documentation and knowledge management of design automation systems. A solution is to be developed for industrial case with required functionality for capturing, structuring, searching, retrieving, viewing, and editing the system embedded information and knowledge. Based on the evaluation of various approaches, an appropriate application will be selected as a solution and implemented according to the relative method. The final solution will be tested based on the preset functional requirements and refined for elimination of problems occurred during the tests. The final solution will enable and facilitate system maintenance, updating and support the reuse of functions and system encapsulated generic product family descriptions.

1.1 Background

As design automation system increasingly applied on customized product, there is massive information and knowledge included in DA system. It is difficult to understand for new users how the system works and how the knowledge is related to each other. This kind of documentation system will help users to understand design automation system that includes the specifications, parameters and rules, also the relationships among them. This will enable the company’s data systematization, classification and integration.

1.1.1 Background of R&D

For many products, the adaptation to customer specifications is essential and requires flexible product design and manufacturing while maintaining competitive pricing by ensuring enhanced producibility (Elgh and Cederfeldt, 2005)[1]. Since customized product strategy is adopted by many companies and in order to enhance competitiveness of such strategy companies themselves, the vision and methodology of automation design system was introduced in, and successfully implemented as well. This technology is not only a means of improved efficiency but also a method for reduced lead times, improved offer precision, quality assurance, performance, and enables a higher degree of customer adaptation.

1.1.2 Theoretical principles

The basic tasks of DA system development are to retrace all variables defining the product and the formalization of the corporate knowledge during design process. The system should be able to adapt with the data increase, diversification and changes of product design over time in the area of product technology, product knowledge, production practice, customer requirements and legalizations and so forth. Hence, for DA system, adaptation of the improved knowledge which governs design is critical and essential. Furthermore, extensiveness and feasibility for different product systems are also the important facts in developing DA system.

The main architecture is depicted in Figure 1-1. Knowledgebase consist several series of files which are executed by different applications while an inference en-
Introduction

gine is used to solve the execution sequence of files. A simple and clear interface achieves user interaction, while the Database is used for connecting enormous knowledge data.

At those point, DSM (Dependency Structure Matrix) is used to present the dependencies between items, and cluster the statements into several different blocks and show the identification of structural levels (Figure 1-2).

DA system realization:

- Theoretically, in the system development process
  - Retrace the required knowledge
  - Close gaps between process and knowledge definitions

- From the system functional perspective
  - Easy to reuse the existing knowledge and information
  - Integrate and automate both design process and quotation activities
  - Speed up the total engineering process with high accuracy result
1.1.3 Introduction of ProcedoStudio

ProcedoStudio is an automatically design software which was developed for KA company, mainly used for automatic design of wiring patterns for car seat heaters. It is used in quotation process to rapid assessment for bids with high accuracy. As further improving, this application will be use in other customized products.

ProcedoStudio is used in Windows system. User interface is programmed by Visual Basic. Application programs which are invoked by ProcedoStudio during the auto-design process include MS Access 2007, MS Excel 2007, MathCAD 13 and CATIA V5R18. Inference engine combines these programs with knowledge objects in Knowledge base to achieve design automatically according to customer specified input parameters and variables. The main structure of ProcedoStudio is shown in Figure 1-3. The knowledgebase comprises rules in CATIA VBA, Excel and MathCAD.

![Figure 1-3 ProcedoStudio System architecture[3]](image)

![Figure 1-4 DSM of DA system[2]](image)
Presently, there are 20 KnowledgeObjects constituting the domain knowledge of designing heating elements. It includes:

- Input 4 pcs.
- Electrical calculations 11 pcs.
- Geometry 2 pcs.
  - Wire layout
  - Amplitude factor
- Manufacturing pcs.
- Cost 1 pcs.
- File handling 2 pcs.

The number of Variables manage by database is 66, although the total number of Variables residing in all of the KnowledgeObjects is of much larger.\[4\]

In the beginning of user interface, the “Available knowledge databases” and “Project” need to be defined (Figure 1-5). When click “new” button, the program will show the customer specified dialog (Figure 1-6). In the user interface, users can click relevant button and add input parameters by selecting needed files in the popup windows. The whole process is visible. Also, user could select different check box to present the order, popup invoking programs and files. After running program, the prototype will be showed in CATIA with 2D draft (Figure 1-7), and all the calculations results will be shown in the window (Figure 1-8).

*Figure 1-5 Database choose of ProcedoStudio*
Introduction

Figure 1-6 customer specified process

Figure 1-5 2D sketch layout

Figure 1-8 calculation results
1.2 Problem definition

In this section, since this project is based on actual case, problems and need of this project will be indicated from both company’s perspective and academic perspective. It will help to define the purpose and research question of the project.

1.2.1 Problems, need and company’s perspective

In the previous project, the design automation system of seat heater has been developed by School of engineering, Jönköping. It provides a significant reduction on lead-time and brought efficiency in doing design work and quotation process to the company (KA). From the company’s perspective, the fact that the system is running with good condition for a long period can be expected. During the practical use of the system in the company, the lack of assistance from either documentation work about the whole system or management of knowledge could bring out some obstacles when the engineers from the company reuse existing knowledge and information without some helps from original developers. As a result, it will significantly delay company’s developing activities, and engineers would be faced great challenges to finish with design tasks and quotation process. Hence, be able to bring full comprehension of both overview of entire system and detailed relationships between different knowledge and information to the users from the company, it is an essential task that explore how to deal with the mapping between scattered knowledge and information which are presented in Figure 1-9.

![Figure 1-9 Knowledge representation of entire system (by Elgh, F.)](image)

From a long-term point of view, reuse of the existing knowledge is a critical issue, since there will be introduction of new product, new variant of existing product, new manufacturing process and additional or modified design rules due to new insights or changes in standards or legislation. In order to deal with these “New” factors, the adaptability of entire system need to be considered, and some support-
ive documentation work and management of knowledge need to be accomplished for achieving the goal of maintenance of the system, updating and reuse of system embedded information and knowledge.

### 1.2.2 Academic perspective

Nowadays, despite many studies of knowledge acquisition and storage for setting up design automation systems, [5, 6] the aspect of maintaining and updating such systems has received rather little attention in the academic world, while companies have faced the practical problem for a long time.

A principle development process is illustrated in Figure 1-10.

*Figure 1-10 Company Development process, information and repositories[4]*

There will be many documents, files and items created and stored in different repositories during the development process at the company. It is vital that make connections between these existing knowledge and information with a good manner which are located in different repositories in order to get the traceability of objective knowledge.

In the product development process knowledge are processed in different steps and appears in different states. Six knowledge states can be identified and these are: unprocessed, selected, structured, adapted, utilized and documented (Figure 1-11). [4] To support reuse, expansion and maintenance of different Knowledge Objects (building blocks) it is required that the focus in the product development process is not limited to the definition of Knowledge Objects exclusively, but also include the definition and collection of associated Meta-Knowledge[4]. Fredrik Elgh pointed out that “Meta-knowledge” can be regard as knowledge that is applied to determine how to execute other knowledge. [4] Comparing with a design automation system, human designers have a higher degree of flexibility on decision making based on e.g. experience, rules of thumb or numerical calculations,
and also on selecting appropriate method or solution based on comparison between different solutions. This kind of ability is expected to be introduced in design automation system for achieving a better ideal solution. Although, the concept of meta-knowledge is well established in the AI world, but its application in practical engineering design problems needs to be adapted and further developed.

![Figure 1-11 Knowledge States [4]](image)

Since the design automation system is developed based on KBE (Knowledge Based Engineering) applications to some extent, there are some issues need to be notified.

**Transparency of KBE applications:**

In order to make more user-friendly, flexible and adaptable knowledge bases, the black-box applications should be moved away. The approach to this purpose is by using sustainable knowledge in KBE, which means the underlying knowledge and supporting documentation for the KBE application should be directly accessible. Another important aspect to level up the transparency of KBE application is use management of the evolution of application. This might be achieved by using more advanced knowledge management techniques and features such as knowledge life-cycle management. [7]

**Enriching the semantics of knowledge models and their traceability in KBE implementation:**

Achieving interoperability enables “data connection” between heterogeneous systems to be used into KBE tools and vice versa. Further exploitation of data entities in KBE automation tools requires a “knowledge connection” where not only the syntactic content of the data but its common meaning is exchanged, an essential principle of the so-called “Semantic Web”. [7]

Documentation is mainly focused on describing the final results of different activities answering “What?” questions. For example, to reuse a rule in a new context (another product family) requires more information, such as scope, range, simplifications and underlying assumptions. Such information might be enough if the rule is to be used as it is, but if the rule has to be modified and adapted to specific circumstances even more information is required to support the adaptation while ensuring the validity of the rule [4].
This thesis project will be undergoing with more detailed questions (pointed out by Elgh, F.) concerning the knowledge management which are listed below:

- How can the executable design rules be documented, structured, validated and made traceable to their associated design knowledge?
- What “meta knowledge” strategies are relevant for applications in engineering design?
- How can the scattered knowledge be connected with reasonable manner that make them traceable from both backward and forward directions?

Some common facts need to be considered, they are:

- The current case is not the only target of the potential solution that will be developed in the future.
- The coming solution should be established on both industrial and academic base, which means the solution will be tested and evaluated for the practical applicability at the company. Whilst from an academic point of view, the solution should be supported by correct and rigorous theories.

1.3 Purpose and research questions

It is essential that for both system longevity and carry-over when developing new product families and the documentation of the system embedded information and knowledge for maintenance, updating and reuse. The objective with this project is to explore approaches for documentation and knowledge management of design automation systems. A solution is to be developed for the industrial case with required functionality for capturing, structuring, searching, retrieving, viewing, and editing the system embedded information and knowledge.

The solution will enable and facilitate system maintenance, updating and support the reuse of functions and system encapsulated generic product family descriptions.

1.4 Delimitations

The main task of this thesis is about documentation management. It will focus on working with managing with support document of DA system to show how it works or relate to each other. Supporting documents here include CATIA, EXCEL, and MathCAD files which will be used during the automation design process, and also instruction file for showing the relationship clearly. This thesis does not include developing the problem itself and the huge data base behind the program (ACCESS). Further, the document used during the auto-design process will also not be revised.
1.5 Outline

Problem definition is set after a general introduction and analysis of Procedo-Studio and seat heater. The related theoretical background detailed in the chapter of “frame of reference”. General Principles and Applications have been compared with the preset functional requirements to select a best option. Structuring Knowledge and implementation of MediaWiki are explained in the chapter of “Development and implementation” as well as test and refine process. The discussion, conclusion and future works have been indicated in the last chapter briefly.
2 Frame of Reference

The purpose of this project is to explore a reasonable approach to represent the knowledge and information underlying DA system. A brief overview and introduction of related supporting theoretical domains need to be reviewed. They are: product development, engineering design, design automation, knowledge-based engineering, design rational and documentation.

2.1 Product development

Product development includes all the activities when companies developing a new product with good performance, low product cost, less development time and minimum development cost. Ulrich and Eppinger defined product development as “the sequence of steps or activities that an enterprise employs to conceive, design and commercialize a product” [8]. Kotler al. (1999) pointed out the new product developing process can be divided into nine steps, as indicated in Figure 2-1. [9]

![Figure 2-1 Nine steps of developing process [9]](image)

Recently, due to dramatic increasing globalization and customer’s extensive demands on product value, effective product development process is becoming more and more vital issue from the company business strategy. According to Smith and Reinertsen (1998), the four key product development objectives are: the right performance for the right product cost within the right development time at minimum development cost. [10] Product development process comprise highly creative and knowledge-intensive tasks that involve extensive information exchange and communication among geographically distributed teams. [11] During the product development process, large amount of, variety of information and knowledge are generated by the engineers and designers. It is essential challenge that ideally deal with these product related knowledge in order to improve the efficiency of the development process.

2.2 Engineering Design

Design is development and realization of something new in systematic way. The result may be a product, a physical object manufactured to satisfy a need while fulfilling all the initial requirements set by different groups of interested parties. [10] In the context of product development, the core of engineering design is the activities within the product design that are accomplished through systematic
application of engineering principles and physical and mathematical concepts to solve problems.[10]

The main four phases in engineering design outlined by Pahl and Beitz (1996) are [10, 12]:

Planning and clarification of the task:
- Starts with an analysis of the market and the company’s situation
- Result in a product idea and a requirement list.

Conceptual design:
- Identification of essential problems
- Establishment of functions structures and solutions that are combined into solutions
- Evaluation and selection of concept

Embodiment design:
- Creation of preliminary layout designs including materials and calculations
- Refining, improving and evaluating
- Results in a preliminary layout that is processed into a definitive layout

Detail design:
- The arrangement, form, dimensions and surface properties of individual parts are set
- The output from this phase is the specification of production

Engineering design is can be considered as a problem-solving process, it comprises a lot of successive decisions in many possible solutions. The systematic approach that making right decision from the initial problem definition to the final solution need to be applied during the whole development process. Meanwhile, it is important that evaluating the feasibility of the possible solutions according to right information, in order to make a correct direction on reaching final goal of development.
2.3 Design automation

Nowadays, company expense a lot of effort, resources and time on dealing with product designing tasks and quotation process. Design automation system combines computer technology with engineering design for enabling efficient design process. There are many redesign activities of an existing product in the company, and according to Encanacao et al. (1990) perhaps more than 90% of industrial design activity is based on variant design. [10] “Engineering IT-support by implementation of information and knowledge in solutions, tools or systems, that are pre-planned for reuse and support the progress of the design process. The scope of the definition encompasses computerized automation of tasks that directly or indirectly are related to the design process in the range of individual components to complete products”. [1]

Elgh, F. pointed out design automation can be divided into two types: information handling and knowledge processing. [10] For instance, an archiving system for the reuse of CAD-files and spread sheet of calculation of a prismatic object and so forth. There also have large amount of knowledge such as rules and algorithms for variant design which are incorporated with a PDM system.

According to Elgh, F., the aim of design automation is to support one or more of the following areas: [10]

**Design synthesis:** includes computerized templates for calculation/optimization of design parameters, applications for calculation/optimization and generation of product geometry, applications that ensure producibility, database systems supporting the reuse of previous solutions, information systems for requirements or manufacturing constraints, configuration systems, etc.

**Design analysis:** by, for example, automated: finite element analysis, geometry preparation for finite element analysis, evaluation of producibility, cost estimation, etc. based on a geometry description and/or design characteristics.

**Plan for manufacture:** comprise computer-aided process planning for the generation of, for example: operation sequences, production parameters, machine control commands, fixture and jig designs, etc. based on a geometry description and/or design characteristics.

The main objective of design automation is achieve the effective and efficient product development process by reducing costs, cutting lead-time, improving product performance and adapting product to different customer specifications.
2.4 Knowledge-based engineering (KBE)

Knowledge-Based Engineering is a research field that studies methodologies and technologies for capture and reuse of product, process and engineering knowledge. The objective of KBE is to reduce time and cost of product development, which is primarily achieved through automation of repetitive design tasks while capturing, retaining and reusing design knowledge.[13] Many companies nowadays, increasingly moving towards mass customization and this requires relatively rapid and collaborative design process and producing process. KBE has its roots in the 1980s, when the application of Artificial Intelligence (AI) and Knowledge Engineering techniques in Computer Aided Design (CAD) became known as Knowledge-Based Engineering.[14, 15] KBE is mainly applied for automating repetitive design process and this is dramatically cutting the lead-time which enabling designers devote more effort on the creative and various design tasks. A major advantage in adopting KBE is presented in Figure 2-2 below.

![Figure 2-2 Achievable design time allocation by KBE adoption](image)

There are several KBE methodologies available to support the development of KBE applications and systems, such as methodology and software oriented to KBE applications, MOKA methodology and KOMPRESSA methodology. [13] A number of research challenges in knowledge-based engineering have been addressed in the conference paper [13], they are:

- Improve methodological support for KBE.
- Moving beyond black-box KBE applications: Transparency of KBE applications; enriching the semantics of knowledge models and their traceability in KBE implementation.
- Effectively sourcing and re-using knowledge.
- KBE success metrics.
- Assessment framework for KBE opportunities.
2.5 Design rational and documentation

Design rationale is an explanation of why an artifact or some part of an artifact, is design the way it is [17]. The processes of maintain and redesign require much effort to understand previous work in the whole product life cycle. Design rationale does a crucial assistance of structuring design problems, presenting related principles to explore more design options. Design rationale system be used as a basis to discuss and reason among collaborating designers which including all back ground knowledge, such as deliberating, reasoning, trade-off and decision making in the design process of an artifact-information that can be valuable, even critical, to various people who deal with the artifact[18].

A design rationale system which is constructed by domain-specific knowledge is used as a representation framework for designers discussing and reasoning. The first approach to design rationale was proposed in 1970 [19], after that, various methods have been proposed and developed, and the exploration never ends. There are two main approaches to developing design rationale systems: process-oriented and feature-oriented. Feature-oriented approach focus on the representation of artifacts and established rules governing the design process, while process-oriented approach is often used to create a historical representation of the design process. [18, 20, 21]

Capture of design rationale is critical process. In design rationale system, the information captured is used to answer or will answer or infer the questions raised by designers. In design process design rationale is captured by recording reasoning, decisions, options, trade-off, dependency relations between data and information, etc., and constructing a formal [22] or semi-formal structure[20] so that the design rationale can be used in the decision-making process during design. The design rationale capture process divide into knowledge recording which refer to capturing raw information and design rationale construction focus on organize and store knowledge. Both of these two processes can be implemented by automatic capture and user-intervention.

Generally, design rationale system is supported by knowledge-based systems and CAD/CAM system [18]. It contributes to design progress by provide designers with a knowledge representation framework, as well as tools to capture design rationale, design reasoning and communication during the design process.

The purpose of documentation is to describe software system, and record the relevant process steps of creation, updating, modification and deletion. Consistent, correct and complete documentation of software system is an important vehicle for the maintainer to gain its understanding, to ease its learning and /or relearning processes, and to make the system more maintainable [17]. It is integrated to present and show the architecture of the software system and the process steps which are visible and visualized to people who need to entirely acquaint. Furthermore, the reasons of some software systems outdated are degradation of the documentation system, which made the problems become more complicate and confusion. Consequently, a correct, complete and consistent documentation is a powerful tool for improving users’ efficiency and work quality, and gain success.
2.6 Summary

Several knowledge domains had been briefly introduced for a better understanding of theories concerned with this project. For instance, the theories regarding to product development and engineering design can be the basic theories that need to be reviewed, and the part of knowledge based engineering and design automation brings a short comprehension to the DA system. Since the part of design rationale and documentation can be our implication of potential solution, there could be more exploration needed regarding to this knowledge domains. To conclude, for both better comprehension to this project and exploration of solution, the theories introduced in previous sections in this chapter will be useful.
3 Functional Requirement

Based on the study and realizability of DA system, a series of requirements of the knowledge representing system will be discussed. Accordingly, the specific functions which could realize the requirements will be explored and expounded.

3.1 Definition of requirement

First of all, because of this project is mainly based on the actual case (Design automation system of seat heater from KA), there are a lot of existing knowledge and information such as specifications, CAD model templates, CAD-macros, design rules and regulations, cost estimations and calculations etc. are generated by the designers and engineers during the developing and design process of the design automation system. These knowledge and information are scattered over and stored at different repositories, or might be saved by a specific individual who is responsible for managing the project.

In order to deal with documentation and knowledge management tasks, we should answer the question “What do we totally have in this DA system?” There are four main different sorts of knowledge that are consisted in design automation system. They are the knowledge that generated in product development process, in engineering design process, in data processing and about application system respectively. It is very important task that collecting these knowledge which will be documented and managed.

Another common question is “How are the information and knowledge distributed, related or connected?” For a better understanding and utilization of design automation system, there is significant need to make clear view on principle of design automation system, and also explore and present the interconnections between these information and knowledge to the possible system users. Design rationale is an effective way of capturing this missing part of an integrated representation of design knowledge, and can be defined from a variety of viewpoints.[23] The possible functions of our potential solution will be followed by the question “What knowledge and information should we concern, and how should we manage with that if we want to reuse (system embedded knowledge), maintain and update”.

3.2 Definition of functions

In this section we will explore and define what functions of our potential solution are, and why we need these functions. The main function of our objective solution is enable documentation task and knowledge management task for the design automation system. From a product development perspective, by using our possible solution engineers could indicate their ideas or thinking which are occurred during the product development process, such as planning the development process, manufacturing process and quotation process. From an engineering design perspective, designers also could represent how the optimal designs came out which will allow the designers can easily make changes in future design work. Some extent of representative design rationale on IT area is also helps system maintenance, and knowledge management skills will be needed for a better systematically man-
Managing the existing knowledge which will provide convenience on traceability and system updating activities. The possible function of our potential solution will be indicated by following sections, they are:

**Creating**: Create new project or product family. Adding design rationale

Creating function is the most common function that exists in almost all the computer system that creating new files, new tasks and new specifications and so forth. In our case, it will be the function that creating a new project, for instance, the documentation system is not only working for seat heater design automation system, it also could create another jobs or project for other specific product. The creating function is also responsible for create new product family, and add design rationales for specific design task and so on.

**Locating**: Locate/Collect underlying or supportive files and documents.

Since there are several different software used and served for design automation system, there exist different supportive files that had been generated by designers and engineers such as Microsoft Word files, Excel files, Catia model files, Catia VBA macro files, Visual Basic GUI and Database Access files and so forth. The locating function is to gather these relative files that are stored in different repositories, and make a connection to our objective system, for enabling accessibility to these files. It is an introducing processes that gathering and localizing supportive files to our documentation system.

**Managing**: Document Management and Knowledge Management/Information Management

Managing function could be the most vital function of our documentation system. There might be necessary to divide this task to some extent, because knowledge management is a big and wide area and it always hard to find an appropriate approach to a specific management task. In this section, we intend to classify the management function as two different approaches.

The first function would be aiming to the document management activities. System underlying document management, such as supportive documents, and design rationale documents, recorded documents which is mainly recording the modifications and output variables and historical generated computational files, all of the documents need to be managed, classified with an appropriate way.

The other function is to manage the knowledge and information that are consisted in the product, development process and application process. These knowledge and information will represented by several approaches. For example, design rationale approach, graphical representation, text transformation of computer code, setting up interconnection between different variables and so on. The key point of this function is to capturing the objective information or knowledge correctly and efficiently for reusing, system maintenance and updating activities. The possible approach will be explored and evaluated in the next phase of this project.

**Structuring**: Product structuring (Part, assembly, material etc.); Process structuring: Design process, Quotation process, computational running process; Product family structuring: Input specification, associated variable specification etc.
Structuring, to some extent, its functional meaning is related to the management function. It aims to ideally organize existing knowledge or information, and present a clear and vivid view of knowledge structure and process structure to the users. Based on seat heater design automation system, there would have a significant need for representing product structure (such as part, assembly, material etc.). In the seat heater design automation system, it contains nicely structured product family. For example, input variables, customer specifications, associated design variables, output variables and so on. In our documentation system, if we can also present this sort of structures to bring a better understanding of information to the users and easier the accessibility to objective knowledge.

Process documentation should be pervasive in and through all of the types of development and maintenance chores. Proper control and management of development and maintenance can only be achieved if we have obtained sufficient visibility into the process, its stages and tasks, executing roles, their decisions and motivations, and the results of each individual process task.[24] Design process as KM (Knowledge Management) core: the design process forms a central element of the knowledge management method, either though process templates, product model integration or knowledge-based process support.[25]

In our case, we intend to make a clear view of the design process of seat heater wire, quotation process and computational running process by mapping and structuring different processes. In order to deal with such tasks, the structuring function is vitally needed.

**Access:** Open/Read/Modify/ (Authorization)

There is no doubt that the function of accessibility is the most basic function in such documentation and management system. Opening the files, read the information that contained in the files, and enables modification of some relative variables depends on different circumstances. The issue of setting up some extent of authorization is desired for the company to insurance information security. The individual who is going to make some modifications on the design variables and rules must take full responsibility for such changes.

**Edit:** Text, Graphical Approach, Figure

Editing function is on purpose of representing better descriptions on the product, process and application system. Editing function will include both text format approach and graphical approach which would provide higher visualized, transparent reflection to the possible users. It is the essential function on dealing with design rationale tasks such as either issue based design rationale or process based design rationale.

**Link:** Issues, variables, rules, process, constraints, code-Links to the design rationale
Link function can be considered as the most fundamental function that achieving knowledge traceability. Such as the issues, variables, rules, process, constrains and computer code will be documented and described specifically (consist description of relative issues, variables, constraints and rules etc.) to some extent by the designers in the late of design stages. Building up the links between the sensitive issues, variables and rules to the respect design rationale is to help the users to find the relative knowledge in an efficient way. Another possible, desired function is that construct the links in an advanced, multi-connected level. For instance, links between variable and variables, variable and rules, variable and constraints and so forth.

**Search:** History Design/Project (store, open, view)

As constantly using design automation systems, there would be enormous generated information because of over and over execution. In order to manage this large amount of information, and retrieve the objective information, reuse existing knowledge, historical searching function is becoming a necessity.

**Recording:** Modified Input variable, output, rules, regulations and calculations.

The purpose of documentation however, is not only to describe a software system but also to record all the relevant process steps leading to its creation, update, modification, and deletion.[24]

The recording function is mainly targeting to enables traceability of sensitive modification activities such as input variables, design parameters, rules and design output. Recording would be done by documenting the modifications and system output for each project. It will provide not only some extent of traceability, but also comparability between different test. And some comment from the users or designers that concerning the design knowledge would also be recorded as experience for future design tasks or test activities.
4 General Principles and Applications

According to the requirements and functions identified in Chapter 3, several different suitable principles and applications supporting the solutions have been surveyed in the related field. In this chapter, principles and applications will be introduced briefly and generally. After comprehensively comparing with the alternative solutions, the final approach will be determined.

4.1 Introduction of principles

The principles are the theoretical methods to solve the analogous problem such as knowledge structure, documentation, design rational capturing and so forth. SysML, MOKA, CommonKADS and PVM are the four selected principles introduced as follows:

4.1.1 SysML

SysML is a general-purpose graphical modelling language that supports the analysis, specification, design, verification, and validation of complex systems. These systems may include hardware, software, data, personnel, procedures, facilities, and other elements of man-made and natural systems.[26] The language is intended to help specify and architect systems and specify their components that can then be designed using other domain-specific languages such as UML for software design and VHDL and three-dimensional geometric modelling for hardware design[26].

SysML can represent the following aspects of systems, components, and other entities:[26]

- Structural composition, interconnection, and classification
- Function-based, message-based, and state-based behaviour
- Constraints on the physical and performance properties
- Allocations between behaviour, structure, and constraints
- Requirements and their relationship to other requirements, design elements, and test cases

There are nine different sorts of diagrams in SysML, they are package diagram, requirement diagram, activity diagram, sequence diagram, state machine diagram, use case diagram, block definition diagram, internal block diagram and parametric diagram respectively.

Considering the potential use in our project, some of specific parts will be discussed in this section. In SysML, constraint blocks are to support the construction of parametric models. A constraint block is a special kind of block used to define equations so that they can be reused and interconnected.[26] In constraint blocks a set of parameters are constrained by an expression. A constraint block can be presented on both a block definition diagram and parametric diagram.
SysML includes a generic mechanism for expressing constraints on a system as text expressions that can be applied to any model element.[26] SysML does not provide a built-in constraint language because it was expected that different constraint languages, such as the Object Constraint Language (OCL), Java, or MathML, would be used as appropriate to the domain.[26]

The element that owns the constraint is shown as a symbol with compartments, such as a block, the constraint can be shown in a special compartment labeled constraints. A constraint can also be shown as a note symbol attached to the model element(s) it constrains, with the text of the constraint shown in the body of the note. [26] Figure 4-1 shows examples of the different constraint notations used in SysML.

![Figure 4-1 Example of two notations for showing constraints][26]

The complex constraint blocks on a block definition diagram can be placed hierarchically in order to present the relations, see Figure 4-2.

![Figure 4-2 A hierarchy of constraints on a block definition diagram][26]

A rationale is a SysML model element that can be associated with either a requirement or a relationship between requirements, or any other model element. As the name implies, the rationale is intended to capture the reason for a particular decision. Although rationale is described here for requirements, it can be applied throughout the model to capture the basis for any type of decision. A prob-
lem is depicted like a rationale, but is used to identify a particular problem or issue that needs.

In Figure 4-3, the rationale is presented as text symbol with the keyword «rationale». The specific expression of the rationale is presented under the key word, and also the relative reference could be connected with by a hyperlink. In this particular example, there is a reference to a trade study, T.1.[26] The context for this particular rationale is shown in Figure 4-4.

![Figure 4-3 Example of rationale as depicted on any SysML diagram [26]](image)

A rationale or problem can be attached to any requirements relationship or to the requirement. For example, a rationale or problem can be attached to a satisfy relationship, and refer to an analysis report or trade study that justifies the assertion that a particular design satisfies the requirement. Similarly, the rationale can be used with other relationships such as the derive relationship. A rationale can also be attached to a satisfy relationship that references a test case which verifies the requirement is satisfied.

![Figure 4-4 Example of the rationale attached [26]](image)
4.1.2 MOKA

MOKA stands for:

“Methodology and software tools Oriented to Knowledge based engineering Application”. [5]

The MOKA project was aimed at reducing the amount of up-front investment and risk associated with the development of KBE applications.[5] MOKA is the methodology which is mainly developed to support KBE, the tasks includes: [5]

- Describing a lifecycle process to guide the user in applying the concepts;
- Finding appropriate ways to represent both the product and process knowledge of the application;
- Building a software tool to enable users to implement the methodology.

The lifecycle of a KBE system adopted for use within the MOKA project is shown in Figure 4-5.

![Figure 4-5 KBE lifecycle identified in MOKA [5]](image_url)

Consequently, support of the KBE life cycle is a useful function of the KBE methodology. This resulted in the introduction of the MOKA life cycle.

Two different sorts of knowledge representation layers are used in MOKA. The first is designed to be very user-friendly and to represent the thinking of the engineers in many different ways. It is called the Informal Model, and the knowledge is classified into five types in this model, they are: [5]

- Illustrations-for recording past experience, case histories, anecdotal knowledge;
- Constraints-restrictions on the objects or the attributes of an object;
- Activities-the elements of the design process
- Rules-knowledge that directs choices in the activities;
- Entities-the objects that describe the product
Each knowledge type has a specific template or form. The set of completed forms, called ICARE forms, holds the knowledge description for the KBE application. The set of linked ICARE forms is the minimum for describing a MOKA Informal model. To make visualization of the links easier, MOKA allows for charts to show the way the product is made up in terms of structure and function as well as to illustrate the sequences in the design process (see Figure 4-7). The ICARE forms can provide a useful reference and training guide for product and design process. The example of Informal model shows in Figure 4-6 below.

![Figure 4-6 The ICARE forms used in MOKA Informal Model](image1)

The second layer of representation is the Formal Model. The knowledge engineer takes the knowledge from the linked ICARE forms and converts it into a UML-style of representation. This second level is not intended specifically to be readable by Experts. Its purpose is to prepare the knowledge in a form that is more checkable by a computer and begins to resemble the representation of the knowledge in coded form that might be used on the KBE platforms.

The Formal Model has two key elements: the Product Model and the Design Process Model. The product model is divided into five views that allow the representation to be constructed in a modular fashion. They are: Structure; Function; Behaviour; Representation; Technology. These views are constructed as UML Class Diagrams. MOKA provides meta-models as a container framework for the knowledge that is held in the Informal Model. The meta-models are used as a guide to build knowledge models that fit the particular application.

The Design Process Model is based mainly on Activity Diagrams to show the sequence of the process and the relation of the elements in the product model to the process. The diagram is showed below in Figure 4-8.

![Figure 4-7 Charts help to visualize the links between the ICARE forms](image2)
4.1.3 CommonKADS

CommonKADS is short for Common Knowledge Acquisition and Documentations Structuring and was developed as a respond to the need for a standard for knowledge-based system. It is a set of comprehensive methodology used for structuring and managing knowledge as well as developing the knowledge-intensive system and so forth. The structured approach is based on following principle of knowledge engineering:

- Knowledge engineering is not some kind of “mining from the expert’s head”, but consists of constructing different aspect models of human knowledge. The CommonKADS model suite is a convenient instrument to break down and structure the knowledge-engineering process.

- The knowledge-level principle: in knowledge modelling, first concentrate on the conceptual structure of knowledge, and leave the programming details for later. In CommonKADS approach, the later is the foremost viewpoint.

- Knowledge has a stable internal structure that is analyzable by distinguishing specific knowledge types and roles.

- A knowledge project must be managed by learning from your experiences in a controlled “spiral” way.

- CommonKADS shows the approach of the problem face by Knowledge Engineers by represent the expert knowledge and design specification in form of text or diagrams.

CommonKADS mainly defined six kinds of modules: Organization model, Task model, Agent module, Knowledge model, Communication model and design model (Schreiber 2000). All the information, knowledge and questions need to be organized could relate to the corresponding model. These information in different models could be reused directly in knowledge-intensive system to reduce the re-development process. It treats knowledge that stored in specific format as
resource, and could be organized and managed. As it illustrates in Figure 4-9, CommonKADS divide the knowledge management actions into three main components: agents, business process, and knowledge assets. They are related to each other and the note in the picture also shows the coincidence with modules.

**Figure 4-9 Knowledge management actions [39]**

### 4.1.4 PVM

Since many companies based their business strategy on customization products and their customer quantity and production scales continue increased with the accumulation of time, the concepts and principles of mass customization was introduced. Moreover, a radical revision of overall business in companies was imperative. Two of the central principles of mass customization are that product ranges should be developed on the basis of modules, and that configuration systems should be used to support the tasks involved in the customer-oriented business processes related to the specification of customer specific products [28].
Specification and specification processes

A specification can be defined as a description which can unambiguously transfer needs or intentions from one group of people to another, such as descriptions of requirements of customer, assembling and installing instructions and so forth. Therefore, throughout the life cycle of the product, mounts of different kinds of specifications was generated which were worked out and can be used every time when a new product is produced. For customization, it is necessary to find a generic way to express specification for every different kinds of customized product. The specification processes which is used for activities connected with setting specifications related to specific customer’s needs is introduced.

Configuration system

The concept of configuration system arose during 1980s in connection with the development of a particular form of IT-based knowledge representation known as constraint-based programming. Two definitions of a product configuration system are: [28]

“Software systems that create, use and maintain product models that allow complete definition of all possible product options and variations with a minimum of entries” [Bourke, 1998, p.1].

“A configurator is a software that assists the person in charge of the configuration task. It is composed of a knowledge base that stores the generic model of the product and a set of assistance tools that help the user finding the solution or selecting components” [Aldanondo et al., 2000].

The procedure of building configuration system

There are seven main phases of building configuration system. The main activity, tools and result summarized in Figure 4-10.
### General Principles and Applications

Based on the principle above, several applications such as Design ration editor, PC pack, Semantic MediaWiki and Product Model Manager will be introduced briefly.

#### 4.2 Introduction of applications

<table>
<thead>
<tr>
<th>Phase</th>
<th>Activity</th>
<th>Tools</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Development of specification processes (Chapter 4)</td>
<td>Flow charts, activity chains or IDEF0 Targeting and gap analysis Framework for structuring product knowledge Other tools: SWOT analysis Scenario techniques Cost-benefit analyses Benchmarking Use case diagrams Project management Change management</td>
<td>Characteristics of the most important specification processes Aims and requirements for the individual specification processes Scenarios in the form of descriptions of future specification processes and definitions of the configuration systems which must support the individual specification processes Evaluation/measurement of the individual scenarios' effect Choice of scenario Plan of action and plan for organisation of further work</td>
</tr>
<tr>
<td>2</td>
<td>Analysis of product range (Chapter 6)</td>
<td>Product variant master possibly with associated CRC cards Framework for structuring product knowledge Other tools: Modularity Scenario techniques</td>
<td>Definition of the configuration system's overall content and structure Product variant master</td>
</tr>
<tr>
<td>3</td>
<td>Object-oriented modelling (Chapter 6)</td>
<td>Class diagram with associated CRC cards and other UML diagrams</td>
<td>OOA model (possibly dynamic), User interface, Requirements specification</td>
</tr>
<tr>
<td>4</td>
<td>Object-oriented design (Chapter 6)</td>
<td>Forms of knowledge representation Criteria for choice of software Class diagram with associated CRC cards and other UML diagrams</td>
<td>Choice of configuration software, An adapted OOA model (OCD model), Requirements specification for programming</td>
</tr>
<tr>
<td>5</td>
<td>Programming (Chapter 7)</td>
<td>Programming and test</td>
<td>Configuration software</td>
</tr>
<tr>
<td>6</td>
<td>Implementation</td>
<td>Plan for implementation Training of users of the system Other tools: Change management</td>
<td>User guide and training plan</td>
</tr>
<tr>
<td>7</td>
<td>Maintenance and further development</td>
<td>Measurement methods Plan for organisation of system maintenance</td>
<td>Measurements of the new specification process' performance, Updated OOA model and programme code, People responsible for maintenance and further development</td>
</tr>
</tbody>
</table>

---

**Figure 4-10: The procedure [28]**
4.2.1 Design rational editor

DRed is a new IBIS-based software tool which allows designers to record their design rationale at the time of its generation and deliberation. The design rationale is documented as a graph of node linked with directed arcs. DRed is one of the latest of many derivatives of the venerable IBIS concept. [29] Thus it allows the issues addressed, options considered, plus associated pro and con arguments, to be captured in the form of a directed graph of dependencies. [29] IBIS simply creates a map of the issues or questions being addressed, each linked to alternative proposed answers, which in turn are linked to arguments for or against them.

There are 4 steps introduced for capturing rationale by using of DRed in “Capturing Design Rational”, they are:[29]

- Diagnosing a problem
- Designing a solution
- Completing a standard checklist template
- Communicating the final design and its rationale

DRed can be used to capture both the problems and corresponding solutions. Then the checklist of standard criteria will be a guide for justification of final design. The narrative description of final solution will be presented as a word processed reports that hyperlinked to rationale outline. The Figure 4-11 shows the illustration of 4 steps that we discussed above. The rationales marked with different colour in the figure below and each colour represents different actions of rationale. For instance, red colour stands for the infeasible solution, and yellow for uncertain situation while green meaning for feasible. Every rationale either regarding problem or solution linked for building a tree structure, and main objective placed at top side of structure.

![Figure 4-11 Illustration of 4 steps in DRed [29]](image_url)
The features of DRed include: based on well-established IBIS principles; easy to learn and use; and robust and flexible.\[29\] In practice it has been found to: help designers view, clarify and structure their design thinking; assist with managing design tasks; capture design rationales as they are created without an additional overhead; and reduce the need for written reports.\[29\]

### 4.2.2 PC Pack

PCPACK5 is a one of the most powerful knowledge management tools which is designed to support retention, sharing and re-use of knowledge in many organizations. By supporting various methodologies such as CommonKADS, MOKA, 47-Steps and SPEDE, it shows high flexibility on both knowledge management and knowledge engineering activities.\[32\]

Whether capturing knowledge to produce an intranet site, a knowledge repository or a knowledge-based system, the tools support a number of key activities:\[32\]

- Analysing knowledge from text documents
- Structuring knowledge using various knowledge models
- Acquiring and validating knowledge from experts
- Publishing and implementing the captured knowledge
- Re-using knowledge across different domains

Such activities are the most common, but critical parts of Knowledge Engineering and Knowledge management projects.\[32\] The use of software to support such activities helps in many ways to make the process more efficient, effective and consistent.

The main functions of PCPACK are capturing knowledge, structuring knowledge, validating knowledge, sharing knowledge and re-using knowledge.

Several tools can be used for helping users to structure knowledge. They are:\[32\]

- Ladder Tool
- Diagram Tool
- Matrix Tool
- Annotation Tool
- Protocol Tool
- Publisher Tool
- Diagram Template Tool

The Ladder Tool enables the user for building various hierarchies of knowledge. The Matrix tool allows two types of matrix (aka grid) to be created and edited: an Attribute Matrix and a Relationship Matrix. The PCPACK Annotation Tool brings flexibility in expression of knowledge objects. The tool allows a page of information to be created and edited for each knowledge object (e.g. concept, attribute, tasks). The Diagram Tool is used for generating diagrams and most sorts
of diagrams can be created. The Protocol Tool is used analysing text based information such as specifications, reports or manuals. The Publisher Tool is used to create websites based on relative information and knowledge, and it provides users can access the knowledge without using of PCPACK. The Diagram Template Tool is used generation of diagram template which is used for defining diagram format. The examples of tools application are illustrated in Figure 4-12.

![Example of various tools used]

**4.2.3 Semantic MediaWiki**

MediaWiki, first launched in 2001, is a free web-based application which is written in PHP program language and uses a backend up database. It is used to run the Foudation’s projects, such as Wikipedia, Wiktionary and Wikinews. So far, Tens of thousands of wikis use it support their websites. Furthermore it has been widely applied to other areas, such as internal knowledge management, group project cooperation assignment assistance and .etc. The software is highly customizable, with more than 768 configuration settings [33] and more than 1,800 extensions available for enabling various features to be added or changed [34]. Hundreds of automated and semi-automated tools have been developed for editing MediaWiki sites. Semantic MediaWiki is one of the extensions to Media Wiki which makes Wiki working as a collaborative and compatibility database by allowing for annotating semantic data within wiki text. Accordingly, comparing with the traditional wiki, Semantic MediaWiki enables to make the knowledge “computer processable” so that greatly simplify the structure of wiki, help users to find more information in less time, and improve the overall quality and consistency of the wiki.

Furthermore, the data structure provide by Semantic MediaWiki is also used to support a large number of extensions. The most common ones are semantic forms, halo, semantic drilldown, semantic compound queries, semantic result formats and semantic maps. Semantic MediaWiki is in use on over 300 public ac-
tive wikis around the world, in addition to an unknown number of private wikis[35, 36].

The benefits showed below:

- Automatically-generated lists: the list in Semantic MediaWiki can generate automatically and up to date.
- Visual display of information: display information by various format way.
- Improved data structure: use simple semantic data instead of categories to reduce classification; easy to manage data by using Semantic templates and Semantic Forms extension.
- Searching information: customize the queries individually.
- Inter-language consistency: resolve data inconsistencies caused by redundant data due to various languages.
- External reuse: realize the external data source invoking in SMW-based wiki.
- Integrate and mash-up data: integrate external and SMW-based data.

### 4.2.4 Product model manager

Product model manager (PMM) is developed for managing product data and information such as configuration, bill of material and the so on. “Product Variant Master”, “Class Information Cards” and “Product Explorer” is the three main modules constitute PMM. It adopts tree view structure to present product configuration visualized in “Product Variant Master” (Figure 4-13). Taking the configuration as a basis, more information and data of the product is combined with relevant branch. Users just need to drag and drop the element from “Module Element”. The basic structure is built by “class” element (root class, part-of class and kind-of class), while the detail information can be added in the same way also, such as attribute, rule, comment, picture etc.

![Figure 4-13 Workspace of PMM [37]](image-url)
Another working module, Class Information Card, is consisted of different folders (Basic information, Relationships, Product Knowledge, Attributes and Values, Rules, Comments, Pictures, Implementation details) which show the information of different class in “Product Variant Master” sheet and the relationships between them as well. More details and information for rules, attributes and so forth could be represent, built, edited and revised in the Cards. See Figure 4-14. The Product Explorer view is mainly intended to be used for navigation between Class Information Cards. Furthermore, the Product Explorer view allows fast and flexible building of models by supporting drag-and-drop and context menus for insertion of elements. (Figure 4-14)

By using Product Model Manager a wide range of benefits can be achieved [38].
- Improved internal communication
- Fewer errors in product specifications (BOMs, quotes, etc.)
- Less resources needed for documentation of product knowledge
- Ensuring that knowledge stays in the organization
- Reduced lead times for the creation of product specifications
- Reduced costs for product specification
- Can be used with minimal or no training
- Provides standard interface for data exchange
4.3 Summary

Several principles were expressed in the section above, and each principle has its own characteristic on presenting knowledge and information. Although basic approaches of knowledge modeling are different from each other, most of the knowledge representation in each model is text based or graph based. This commonality between different principles could be a breaking point, and with combining an appropriate application on that, a solution could be coming out in next phase of project.

4.4 Ideas of alternative approach

It might be hard to determine which principle is the best option for this case as well as the using application due to the specificity of documentation on the DA system. According to the general principles that were listed, the rationale presentation used in SysML could be used in our case, while the approach of system modeling used in CommonKADS can be our theoretical support. As well as MOKA declared the systematical way of knowledge representation for both product and process. Since each of principle has their own advantages in presenting knowledge, we can highlight the excellence of each principle analyzing the possibilities of mutual integration of various principles. If the approach works, then an application tool would be needed for supporting the new method.

4.5 Evaluation and selection of approach

DRed has its own method of recording design rationale at the time of its generation. It is an IBIS-based approach, and considering the DA system includes massive computational knowledge like programming code, DRed could not be our solution.

PC PACK could be a possible solution because it can support various methodologies such as MOKA and CommonKADS. It also has powerful tool to do the documentation work and this was mentioned specifically in the previous section. It is commercial software which is used in many industries and its practicality has been proved by many companies around the world.

Product Model Manager is a user-friendly tool that mainly used for documentation of product related knowledge. Due to lack of functions of representing the knowledge concerning the developing process, this application can be out of our consideration.

Semantic MediaWiki is a web based publishing tool with extraordinary powerful text representative function. It also has an attractive function that easy to set up the connection between different knowledge by using of hyperlinks. Although MediaWiki does not have the knowledge modeling or structuring knowledge tools, it has excellent representative performance. Which means any kind of knowledge expression can be presented on the MediaWiki web and it also enables the sharing of these knowledge and information among a group. Finally, considering the high flexibility of using MediaWiki, We decide to consider it as our final selection of application.
5 Development and Implementation

In this section, we are going to talk about the implementation of MediaWiki, it includes collecting knowledge and information behind the DA system; Structuring the knowledge; and the specific application of MediaWiki. Since we have decided to use MediaWiki as our final option, there is a need to explain what sorts of benefits that we can obtain from using MediaWiki, and what profitable functions does it have. As we have mentioned in previous section, MediaWiki is a powerful, effective and convenient tool on doing texting and editing work. It basically used web based editing environment, it has a lot of useful editing functions and view functions, which means the users not only have a high flexibility on text representation, and also have a clear view site. One of the most remarkable functionalities of MediaWiki is easy to set the hyperlinks for a specific word or sentence. It will help us to build up the interconnections in a simple way, for example, the interconnections between rules, calculations, specifications, tasks, processes and even a couple of variables could be set up in a highly efficient way.

5.1 Collecting knowledge and Information

Collecting knowledge and information is one of the most important steps on doing documentation of DA system. At first, we need to figure out what kinds of knowledge do we have totally in DA system. Basically, the DA system has been developed and supported by large amount of knowledge which is filed in a different way with its unique role to the whole system. These supportive files are Design Rules, Specification, Type Geometry, Electric Calculation, Design Process, CAD preparation, Macro modules, Macro code, Macro documentation, Production Preparation and Cost estimation, See Figure 5-1.

![Figure 5-1 some knowledge object in DA system](by Elgh, F.)
Each file has its own form, working environment and functionalities, and it also has the author of the file and filed date. For example, the design rules were documented as MicroSoft Word form, while the specifications for calculating task are applied as MathCAD application, See Figure 5-2.

<table>
<thead>
<tr>
<th>Object</th>
<th>Base Form</th>
<th>Form</th>
<th>Author</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harnes Resistance and</td>
<td>Microsoft Word</td>
<td></td>
<td>Johnson</td>
<td>2009/9/2</td>
<td>for winding pattern on heating element(K1.1-K1.4)</td>
</tr>
<tr>
<td>Heating Wire Selection</td>
<td>MathCAD</td>
<td></td>
<td>Fridek</td>
<td>2009/9/30</td>
<td>Selection of Heating Wire: The selection of Heating Wire gives the Wire Resistance and Allowable Wire Diameter</td>
</tr>
</tbody>
</table>
based approach to structure the knowledge and information. The process is meaning of computational system-running process, see Figure 5-3 (running process).

Figure 5-3 System process flow from Analysed DSM [by Elgh, F.]

It shows two different system-running processes which are called Design Sequence Geometry and Design Sequence Electric with a number of tasks. The tasks in this process chart were set based on customer requirements and design rules, and the sequence of process is depended on analyzed DSM, see Figure 5-4.

Figure 5-4 Analysed DSM [by Elgh, F.]
The basic idea of structuring knowledge of DA system is that set the different layer of knowledge object in order to get both clear view of division and hierarchy of knowledge object. For example, the process layer is the top layer which is the main root of system, and the tasks in the process are set as the second layer. Then under the task layer, there is several parallelized knowledge object including Rules, Electric Calculations, Specifications, Macro Modules etc., see Figure 5-5 (Hierarchy figure).

![Figure 5-5 Basic Hierarchy](image)

The idea of this approach is simple and reasonable. In the electric sequence, the rules were set by customer and electric calculation, specification and geometric generation were designed by engineer. A certain extent of calculations, specifications and rules are involved in a manner of responsibility to solve a task. It means that these knowledge objects are all interconnected and cooperated with each other to solve tasks. In the geometry sequence, tasks are realized by running the program. Other knowledge objects can relate to the task directly or indirectly with the bridge of procedure. The key point of this approach is exploring the relation between different layers, and also between different knowledge object in the same layer. By setting up the connections, the users can more easily access the knowledge that they want and understand the total system in a simple way. The indication of how to build the interconnections will be introduced in below.

### 5.2.1 Design sequence electric

This process is the automatic calculation process based on customer specifications and rules, and the process flow is determined by analyzed DSM. In this calculation process, each task has the related rules, specifications and electric calculation. The rules were directly set by the customers, the specifications and electric calculations were set by the engineers by following predetermined rules. The rules are indicated as text form in MicroSoft Word files, while the others are MathCAD files which can be used for system automation, see Figure 5-6.
As it was mentioned previously, the tasks in the process have some extent of connections to the rules, specifications and electric calculations. In the some cases, the tasks are only related to the specifications and rules, while some tasks are related to all of them. The main purpose of exploring the relations is to build up interconnections between these knowledge objects. See Figure 5-7.

Figure 5-7 Interconnections between different knowledge objects

Figure 5-8 below shows the name and abbreviation of tasks, rules, electric calculations and specifications which are related to this process.
Two examples of setting interconnections will be taken in next phase in order to describe the connecting process specifically. At first, we take Task 20 as a starting point. Figure 5-9 shows that, the name of Task 20 is “Harness Resistance”, and we can easily find a rule also called “Harness Resistance” from the rules document with its relative description and formulation. Here the connection between tasks and rules has been developed. By going through computational knowledge, there is an electric calculation which was filed as MathCAD form had a name of “Harness Resistance” and it is abridged as “EC4”. The purpose of calculation, input parameters, output parameters and operations are all introduced in this MathCAD file. In EC4, there are two input variables—“Harness Resistance” and “Harness Length” with output of “Harness total Resistance”. Then there is a need of tracing the origin of these input variables, and these input variables were one of the outputs of Specification which is called “Associated Design Parameters”. Consequently, the connection between the different knowledge objects has been found. See Figure 5-9.
An interesting point in this case is that there exists a relationship between the rules. “Rule 1.2” is the main rule, while “Rule 1.1” plays a parent role of “Rule 1.2” and “Rule 1.3” is a child of “Rule 1.1”. With this approach, the relationship between different rules can be set up, and it is also possible to have a “Sibling” relationship with other rules. See Figure 5-10.

The starting point will be changed in another case. By beginning with the “Rule 3.4”, we can easily find that it is for solving “Task20”, and content of the rule is indicated in “EC13”. The principle of finding relations between these objects is same as the first example. The reason of taking this example is to confirm that the connections between those knowledge objects have capability of both “top-down” and “bottom-up” searching. There exists the relationship between different electric calculation as same as the rules have it. Basically, the connections can be found from the relations between input variables and output variables in the electric calculation. As a result, in this case, “EC13” has the Parent-child connection.
with “EC4”, “EC11” and “EC2”, while “EC2”, “EC11” and “EC4” have sibling connections. See Figure 5-11.

![Figure 5-11 Hierarchy relations of Electric Calculation](image)

By following this approach, the relationship between the knowledge objects in “Design Electric Sequence” had been set up. The result of this will be showed in below table and diagrams. The Figure 5-12 shows the relationship between different knowledge objects, and according to the table we can depict diagrams that indicate the relations R (Rules) to R, EC (Electric Calculation) to EC, and EC to S (Specifications). See Figure 5-13.

<table>
<thead>
<tr>
<th>Task</th>
<th>Related Rules</th>
<th>Electric Calculate</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>T8</td>
<td>R1.1 (main)</td>
<td>EC1</td>
<td>S1</td>
</tr>
<tr>
<td></td>
<td>▲R1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T10</td>
<td>R3.1 (main)</td>
<td>E07</td>
<td>R3, S3</td>
</tr>
<tr>
<td>T19</td>
<td>R3.2 (main)</td>
<td>▲E01, EC10</td>
<td>R3, S3</td>
</tr>
<tr>
<td></td>
<td>▲R3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T20</td>
<td>R3.1 (main)</td>
<td>E04</td>
<td>S1</td>
</tr>
<tr>
<td></td>
<td>▲R3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T23</td>
<td>R3.1 (main)</td>
<td>E01 (main)</td>
<td>R2</td>
</tr>
<tr>
<td>T25</td>
<td>R3.3 (main)</td>
<td>▲E01, EC11</td>
<td>R1</td>
</tr>
<tr>
<td></td>
<td>▲R3.1, R3.2</td>
<td>▲E01, EC1, EC7, EC10</td>
<td></td>
</tr>
<tr>
<td>T26</td>
<td>R3.6 (main)</td>
<td>E02 (main)</td>
<td>R3</td>
</tr>
<tr>
<td></td>
<td>▲R3.4</td>
<td>▲E01</td>
<td></td>
</tr>
<tr>
<td>T27</td>
<td>R3.4 (main)</td>
<td>E02 (main)</td>
<td>S1</td>
</tr>
<tr>
<td></td>
<td>▲R3.2, R3.5,</td>
<td>▲E01, E04, E01</td>
<td></td>
</tr>
<tr>
<td>T36</td>
<td>R3.5 (main)</td>
<td>E011 (main)</td>
<td>R1</td>
</tr>
<tr>
<td></td>
<td>▲R3.2</td>
<td>▲E01, E01, EC11, EC2</td>
<td></td>
</tr>
<tr>
<td>T37</td>
<td>R3.3 (main)</td>
<td>▲E01, EC13</td>
<td>R1</td>
</tr>
<tr>
<td></td>
<td>▲R3.2</td>
<td>▲E01, E04, EC11, EC2</td>
<td></td>
</tr>
<tr>
<td>T38</td>
<td>R4.2 (main)</td>
<td>E06 (main)</td>
<td>R1</td>
</tr>
<tr>
<td></td>
<td>▲R3.6</td>
<td>▲E01, E04, EC11, EC2</td>
<td></td>
</tr>
<tr>
<td>T41</td>
<td>R4.4 (main)</td>
<td>E05 (main)</td>
<td>R1</td>
</tr>
<tr>
<td></td>
<td>▲R4.1, R4.2</td>
<td>▲E01, E04, E06</td>
<td></td>
</tr>
</tbody>
</table>

![Figure 5-12 Total Interconnection](image)

**5.2.2 Design sequence geometry**

The other sequence of DSM is geometry part which runs the CAD-model in CATIA by invoking the parameters which are calculated by the electric sequence to obtain the initial 2D sketch of wires. In the Knowledge representation of the...
entire system, Macro documentation, type geometries, CAD-preparation, Macro modules and Macro code has been bound up with the geometry sequence. The main purpose of this section is to present all the knowledge of the design automation system and build up the interconnections between different elements in a specific knowledge object or between different knowledge objects.

### 5.2.2.1 Knowledge presentation

Due to the form and content of the knowledge objects in geometry sequence are extremely diversified, there is need of relative explanation regarding to several knowledge objects before structuring knowledge.

#### CAD preparation

To run the design automation system, the first step is to prepare CAD model based on solid model of actual pattern carrier. The specific steps are quite clear and easy to understand which detailed in “ProcedoStudio.NET User Manual”. This enables customers just need to follow the instruction step by step to create new models. Thus, the method of direct text based description was taken for this part. The instruction is shown in Figure 5-14 as follow. In addition, the information of “where is it from”, “what is it used for” etc. needed to be presented in MediaWiki.

![Figure 5-14 Instruction of building up CAD module](image)

#### Type geometries

As we mentioned before, a certain structure need to be built in the 2D contour sketch before it could be used in ProcedoStudio by using “Geometrical template” file. It involves a number of pre-determined geometrical parameters. This geometric template as a basic frame of parameters is used over and over again during system running. On the left side of the CATIA user interface of the template file, the structure and parameters is shown as a treeview navigation, see Figure 5-15. The
parameters set in the template are same as they are in Macro. The minimum hierarchy of the model file is shown in Figure 5-16. The type geometries will presented as a basic knowledge by Media Wiki.

Data structure

In this object, data structures and types in the program are generally shown to help user understand the code. It organized the date type in to several classes such as “Collector Organization”, “Specification Parameters”, “Design Parameters”, etc. and showed the information by tables with brief description. Figure 5-17 is one of the forms present the members in “CollectorHierarchy”.

<table>
<thead>
<tr>
<th>Member</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part</td>
<td>Part</td>
<td>The top-level part within which all geometry is placed</td>
</tr>
<tr>
<td>Pattern</td>
<td>OrderedGeometricalSet</td>
<td>The collector that holds the final assembled geometries</td>
</tr>
<tr>
<td>Elements</td>
<td>OrderedGeometricalSet</td>
<td>The container in which the starting geometry is situated</td>
</tr>
<tr>
<td>Temp Offset</td>
<td>Body</td>
<td>Holds all temporary geometry generated at runtime</td>
</tr>
<tr>
<td>Aux</td>
<td>Body</td>
<td>Auxiliary collector in Temp. Pretty much all the supporting geometry are situated here (for example, measurement sketches, pattern line splits, intermediate assemblies, etc.)</td>
</tr>
<tr>
<td>Split</td>
<td>Body</td>
<td>This collector the geometry related to profile splitting — splitting lines and profiles.</td>
</tr>
<tr>
<td>ParamSet</td>
<td>ParameterSet</td>
<td>Points to the root parameter set in the part</td>
</tr>
<tr>
<td>Names</td>
<td>CollectorNames</td>
<td>A structure that contains the name(s) of all the collectors defined here, as they are retrieved from the specification tree by name</td>
</tr>
</tbody>
</table>

Figure 5-17 data structure [30]
Macro modules

Macro modules generated when recording the macros in CATIA application, and these modules belong to different library according to its characteristics and functionalities. The macros in CATIA VBA are consisted by several modules named as “mRoutines”, “mCase”, “mSketch” and “mPatten” etc. which is shown in Figure 5-18, and the detail of each module is indicated in Figure 5-19. Each module plays corresponding functional role in program running. The method for classification of Macro modules also applied in other knowledge objects. For instance, in the function reference, hundreds of functions are stored in different category named “mCase” module, “mSketch” module and so forth. The functions in each container are one-to-one correspondence. All the functions involved in modules need to be described in general, and the programs in each module need to be detailed into macro code knowledge object.

Macro code

The last layer of the knowledge structure is Macro code. Since it is the lowest level of the constructions, almost all of other objects have directly or indirectly relation with it (See Figure 5-20). In the code section, the procedures need to be shown in a clear way for users reading conveniently. Inside of the code and procedures, large amount of invoking actions are used, and such actions are achieved by setting up hyperlinks. These relations will help user locating programs quickly that enables easy and convenient searching, revising and maintenance.
All the geometry sequences are based on the running of CATIA macros with the value of user-specified parameters, the variables gained from the electric sequence and Prepared CAD model. It is initial step in building knowledge structure and doing documentation regarding to the macros. For fully structuring and presenting the knowledge and information, the program flows has been taken as backbone on which presenting other knowledge and information. The following diagram shows the steps and the sequence of macros-running from an algorithmic point of view. As it shows in the picture, the orders of initialization, retrieve specification and create layout are the first three stages (with the color of light blue), see Figure 5-21.
The bulks of the program are part of the last top-level stage—the pattern creation. In these stages, patterns are created with the customer specified parameters and calculated variables together. The functions with red color are meaning of containing sub-branches (Calculate parameters (tier1), calculate pattern parameters (tier2), create return wire, create offset profile, create pattern, create offset profiles, create pattern) will be shown by the approach of hyperlink. Besides, a whole program flow picture including both main backbone and sub-branch will present at the final part where the execution order of the entire process can be clear at a glance (Figure 5-22).

![Figure 5-22 integrated process flow [30]](image)

**Function reference**

Function reference is one of the most important sections to help user understanding the process flow and combining process to programming code. Since it demonstrates all the functions developed in the code, it could be a useful reference in having a deep study about the entire system. The function definitions will follow the outline below approximately. It includes status, arguments, output, description, method visibility; input data specification and so forth (Figure 5-23). Since hundreds of functions are in this reference, we define “Method No.” as the name of each function.

The macros in CATIA VBA consisted by several modules named as “mRoutines”, “mCase”, “mSketch” and “mPatten” etc., and each module has correspondent function, while it contains a number of methods to comply different sub-functions (the detailed information will be introduced in the part of “macro modules”). To take the outline as template, the knowledge and information is uniformly shown in an uncomplicated way. Take one method as an example which is shown in Figure 5-24. The multi-layered function with hyperlinks of MediaWiki should be highly used to present the internal construction and affiliation of methods. Besides, the invoking relationship among methods, modules and other knowledge objects need to be spread as well.
5.2.2.2 Interrelation presentation

In the previous section, the knowledge objects which support running the system and storing information of the system have already introduced. The complicated and interlaced relations among them considered as a whole construction will be organized and established in a reasonable way. A picture (Figure 5-25) was drawn to show the general relationships between different knowledge objects, and by going through the relationships we can easily and quickly figure out the objects that need more focus, while which objects are relatively associated. (Due to the practical DA-system is more complicated than the DSM tasks set in earlier stage as well as it is too difficult to match each method, code and task one-to-one correspondence, so that the relationship between DSM (tasks) will not build here). From this specific project, we can draw inferences to other DA system. In DA system, process flow is used as basis of developing code in earlier stage. The existing knowledge objects of Process Flow and Code are essential, and they are closely related to each other. The codes either generated from macro or are written artificially, they should be organized by a certain way. “Function description” could easily bridge the “Process flow” from “Code”, and help users to understand code deeply. These knowledge and information connected tightly. To make the documentation system understandable, the relatively associated knowledge objects cannot be ignored. Thus, Listing the main knowledge objects and selecting the specific knowledge objects that have mutual relationships can be considered as the first step in setting up knowledge structuring process.
During the developing process of DA system, the tasks in DSM defined in the early stage were unspecific and inadequate. The number of task has been increased and the process flow becomes more complicated than before. Furthermore the connection between tasks and knowledge is not as direct as it was in the Electric sequence. It is extremely hard to use the same approach that takes tasks as the starting point in the process of exploring connections. As we can see, from the relations analyzed above in Figure 5-25, the knowledge objects could be divided into three groups: basic knowledge group (relative associated part), theoretical supporting group and code group. The basic knowledge group presents the knowledge which is needed to be understood by users basically in order to deeply go through the other two parts.

Code group is the procedure in CATIA VBA which will run to create pattern according to the specifications and parameters set. The theoretical supporting group is theory and analysis of the code part. Both code and theoretical supporting groups are the most critical exploring targets through the several knowledge objects in process of structuring knowledge included in the geometry sequence.

As it was wrote in the structure knowledge part, layers inside and among knowledge objects need to be built at first, and then finding the relations in each layer or between different hierarchies. From the analysis above, the first layer of the whole structure consists basic knowledge group, theoretical supporting group and code group. Under the groups mentioned above, a number of corresponding knowledge objects expended in the second layer. (Figure 5-26). This classification method is also applicable in other DA system.
From the third layer, the relations will be built, and this is the most critical parts of geometry sequence. In the third layer, a reasonable and feasible way need to be explored, and the relations can be clearly shown among objects, such as instructions and description, son and parent, call and be called and so forth.

As it was mentioned before, the DSM process set in the earlier stage for DA system is not suitable for Geometry Sequence. But the process based approach had been applied in this part of sequence. To follow this reason, we need to find a basic process which can be used as substitution and more suitable. Since the flow chart is comprehensive, and throughout the whole sequence, it could be taken as a backbone with which the other knowledge and information could be related. It also could be used as a bridge to build indirect relation among knowledge objects. To prove the feasibility of this approach, some part of the relations will be taken for instance in Figure 5-27.
At first, the section marked by label ② was selected as an entry point. According to the name of these steps (Bind to current CATIA document, Create geometry collectors and Create parameter sets) inside the blue blocks, we can easily find the matching functions (numbered as M2 and M3 in label ③) in Function Reference. Thus, the relation between function and process flow have already set up. The second step is to find the connection between Function Reference and Code. According to the information detailed in each function, we can quickly track the procedure ④ in code group. The same classifying approach in “Function Reference” and “Code” helps us locating easily and quickly. It means M2 and M3 are under the class of “mRoutine”. In the code group, the procedure we were searching for must be in the class with same name. The relation between ②, ③and ④ built up so far. Another section is chosen to show the process of building up the connection is marked by label ⑥ in the Process Flow picture. This step in the flow is “Retrieve specification”. Following the way in last instance, we can find the function to support this as well as procedure. The reason to state the second example is all the procedure in “mCase” is one-to-one cooperation with the functions under the class with the same name. From here we could know, one function must have a counterpart of procedure in “Code Group” as well as the steps in “Process Flow”. In addition, the order of build up the relation is not constant. Sometimes, it is easier to find the connection between process and code at first, then turning back to the flow.

Another main part of the relation is inside one object: Code which is classified by different mould. As same as the regular program, inside the procedure, there are lots of calling, invoking among different modules. In order to deal with the task of refining, maintaining and revising of programs more convenient, simple and avoiding of missing, the traceability is the first priority to be considered as a target. The classification of program calling of each other is illustrated in Figure 5-28 below. The program runs from the main module ①. It can invoke an entire module ② in the other layer as well as calling a set of independent code ③ in other modules. The calling relationship also exists between different modules in the same layer ④, while the interlayer calling is used as well ⑤.
5.3 Implementation

Since the knowledge and information included in DA system has been structured, it is necessary to upload the knowledge on MediaWiki web. At first, the main web page is needed to be designed into a reasonable layout. There is a brief introduction concerning this project and DA System at the top the main pages, and this is on purpose of giving understanding to the new users. The short introduction includes the expression about “Procedostudio” which is the main operational platform of Heating Wire Design Automation System, and also there is indication of knowledge representation of DA system. According to knowledge presentation, we set hyperlinks for a group of knowledge objects for the connections with specific knowledge and information that imported at later steps, See Figure 5-29.
From the overview of knowledge objects, by clicking a specific object on the main page, it is possible to jump into another new page due to preset hyperlinks. In this new page, we can freely edit the content of the page. For example, all of rules were listed systematically with its name and abbreviation. For the abbreviation-the substituted name of rules, there also need to set hyperlinks which can jump to another new subpage with expressing specific knowledge corresponding to a specific rule. By following this approach, a triple sub-connection had been built. The contents form had been created automatically by MediaWiki based on the list of rules which brings a clear overview of all the rules. Since the connections had been developed, the specific knowledge from original MicroSoft Word file needed to be presented on a new web. This is a simple process since the original rules’ content has been written as text form, and MediaWiki has competent and excellent functionality in doing such text based editing works, see Figure 5-30.

The situations on the other objects are almost similar to the mentioned above. In the process of importing knowledge from raw files of both electric calculation and specification, the same approach can be used. Since each of knowledge object that from top level to the most specific level had a demonstrative space, then the interconnection between these knowledge objects can be set up easily by using hyperlink function in MediaWiki. In MediaWiki, hyperlinked text has its dependent character, and it has its own space or web page for annotating. For example, once a word is hyperlinked, then it has a new page can do demonstration. Figure 5-31 indicates the specific working principle of interconnections set up in MediaWiki. By following the approach of structuring knowledge that mentioned in previous section, the interconnections had been set up among different knowledge object. In “EC4” page, not only relative information has been revealed, but also relative connections had been set up by hyperlinks at the end of page, they are “R1.2”, “S1” and “T20”. By clicking these hyperlinks, the pages with corresponding specific information will be showed up as the figure shows. More importantly, due to the preset the hyperlinks in the other pages, it is also possible to access to relative pages with a simple clicking.
The approach used in geometry part is also similar to what have done previously in electric part. Hyperlink function in MediaWiki is the most useful and appropriate function in dealing with documentation work. By using this function, any words and any sentences could be annotated specifically, which means massive amount of linkable knowledge can be completely and exactly expressed. In dealing with tasks of geometrical generation, it is a tough challenge on doing documentation work due to large amount of computational knowledge such as programming code, expression of related modules with its massive included functional method, and highly complex processing flow.

The main structure of the knowledge objects present on the “main page” of MediaWiki. According to the works we have already done in the knowledge structure section, the basic construction is illustrated in the Figure 5-32 below. It contains all the knowledge objects with hyperlinks, and the relative page of each knowledge object can be easily access by simply clicking.
The basic process flow of “Geometry Sequence” that substituted from original task process have already related to the functions in “Function Reference”. Here the names of function have been abbreviated as “Method_No.” while the numbers abbreviated into “MNo.”, see Figure 5-33 bellow. Some of the steps in the picture were marked by abbreviation like “mParam_Module” which means it is related to all the functions in class of “mParam Module”.

From the example we used in the “knowledge structure” section previously, the small part marked (2) is related to Method_2 and Method_3. Then back to the main page, by clicking the hyperlink of “Function reference”, we can jump to the subpage of function reference, see picture 5-34. In this page, a short introduction regarding to the function reference will be indicated as well as the function outline. The catalogue of the class for hundreds of methods is beneath the function outline (same as the classification in the raw material). There is also a text to show the Method number in each class. By clicking corresponding link of “mRoutines Module”, a new page will popup which contains all the functions with an automatically generated catalogue, see Figure 5-35. At the end of function name, hyperlinks have been set as the character of “C1” which links a page of revealing corresponding code to its function. By using this kind of approach, the knowledge structure is successfully applied to MediaWiki.
Figure 5-33 Process flow with functions

Figure 5-34 Demonstrative page of Function Reference
The other part of knowledge documentation regarding the Geometry Sequence was presented by the connection between different procedures. Figure 5-27 was shown both the hierarchy and calling relationships of the procedure. All the programming code is stored in the Macro Code page. By clicking the hyperlinks, a new page with showing the details of the code will popup. In some specific cases, the sub-functions included in a main function used for invoking program, and it has its own name of function and code expression. By following this theory, a new relation can be set up by using of hyperlinks in MediaWiki shows in Figure 5-36.
5.4 Test and evaluation

In this section, the function of implemented MediaWiki will be tested by following the comparison to the preset functional requirements in Chapter 3. A simple test approach will be applied by using a form which contains the function’s name, a short description, solution, problem and ideas of refinement. The outline of the template is shown following Figure5-37.

<table>
<thead>
<tr>
<th>Name</th>
<th>Function Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>A short description of preset function.</td>
</tr>
<tr>
<td>Solution</td>
<td>Indicating the way of the function is satisfied by using MediaWiki</td>
</tr>
<tr>
<td>Problem</td>
<td>Declare the shortage or the problem occurred in the solution</td>
</tr>
<tr>
<td>Ideas of Refinement</td>
<td>Pointing out the possible approach of solving the problem.</td>
</tr>
</tbody>
</table>

*Figure 5-37 Template for test expression*

<table>
<thead>
<tr>
<th>Function</th>
<th>Creating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>new project/product family</td>
</tr>
<tr>
<td>Add/change design rational</td>
<td></td>
</tr>
<tr>
<td>Solution</td>
<td>The first function of creating is for creating new project or new product family as well. A new project can be set up by creating a new web and this can be done by applying new account in MediaWiki. In MediaWiki, the set of product family is challengeable, because it does not have the function like building up a tree structure. The function of adding or modification of design rationale can be easily done by editing the web page of MediaWiki.</td>
</tr>
<tr>
<td>Problem</td>
<td>No direct function of building a tree structure;</td>
</tr>
<tr>
<td>Ideas of refinement</td>
<td>The problem can be solved by building a structure of hierarchical hyperlinks regarding the product structure. More importantly, the images of sub-elements can be presented at the end of hyperlinks. See the Figure 5-38.</td>
</tr>
</tbody>
</table>
Development and Implementation

Figure 5-38 Example of setting up tree structure

<table>
<thead>
<tr>
<th>Function</th>
<th>Locating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>locate the specific file/information/document Collect underlying or supportive file/documents</td>
</tr>
<tr>
<td>Solution</td>
<td>The files can be uploaded on the web page, but the type of files is limited by the grade of users. The files have already uploaded on the web can be downloaded. Locating can be achieved by a simple search of file’s name.</td>
</tr>
<tr>
<td>Problem</td>
<td>The types of uploaded files are limited (can only upload “png”, “gif,” “jpg” and “jpeg” currently)</td>
</tr>
<tr>
<td>Ideas of refinement</td>
<td>Reset the limitation in the MediaWiki by extending file’s type.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Managing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>document management (manage, classified) Knowledge and information management Capturing information/knowledge objects</td>
</tr>
<tr>
<td>Solution</td>
<td>Uploaded files totally presented in a specific page in MediaWiki, and by creating a new page for the classification of documents with preset hyperlinks to uploaded files, the managing can be achieved. Knowledge and information that related to the DA system can be easily shifted to and presented on MediaWiki.</td>
</tr>
</tbody>
</table>
A reasonably structured and connected knowledge in MediaWiki enables capturing the needed knowledge, and the searching function in MediaWiki brings the efficiency in the capturing process.

Problem

- 

Ideas of Refinement

- 

<table>
<thead>
<tr>
<th>Function</th>
<th>Structuring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>product structuring (assembling structure, material list) Process structuring (Design process, Quotation process, Computational running Process)</td>
</tr>
<tr>
<td>Solution</td>
<td>The structuring product or process related knowledge had been completed without using of MediaWiki.</td>
</tr>
<tr>
<td>Problem</td>
<td>No specific structuring tool in MediaWiki</td>
</tr>
<tr>
<td>Ideas of refinement</td>
<td>The reasonably structured knowledge had been shifted and clearly presented in MediaWiki. Due to the function of hyperlinks, the interconnections between different knowledge object can be set up.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Authorization of Open/Read/Modify</td>
</tr>
<tr>
<td>Solution</td>
<td>In the MediaWiki web, authorization of accessing to the information can easily set up by administration. This is significant issue since the web should be shared to many users.</td>
</tr>
<tr>
<td>Problem</td>
<td>-</td>
</tr>
<tr>
<td>Ideas of refinement</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Edit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Text, Graphical Approach, Figure</td>
</tr>
<tr>
<td>Solution</td>
<td>MediaWiki performs excellent in text editing, and both creating tables and image presentation are possible.</td>
</tr>
<tr>
<td>Problem</td>
<td>MediaWiki has no specific function for graphic application</td>
</tr>
<tr>
<td>Ideas of refinement</td>
<td>Using of other application to create specific graph, and upload it as an image. Adding the expression of graph and set hyperlink on the key word that presented in the graph.</td>
</tr>
<tr>
<td>Function</td>
<td>Link</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Issues, variables, rules, process, constraints, code-Links to the design rationale</td>
</tr>
<tr>
<td><strong>Solution</strong></td>
<td>The links between different knowledge object has been set up by using of hyperlink function in MediaWiki.</td>
</tr>
<tr>
<td><strong>Problem</strong></td>
<td>-</td>
</tr>
<tr>
<td><strong>Ideas of refinement</strong></td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>History Design/Project (store, open, view)</td>
</tr>
<tr>
<td><strong>Solution</strong></td>
<td>There is a powerful searching function in MediaWiki, the knowledge and information can be found by searching of keyword, and the result shows with “page title match” and “Page text match”.</td>
</tr>
<tr>
<td><strong>Problem</strong></td>
<td>-</td>
</tr>
<tr>
<td><strong>Ideas of refinement</strong></td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Recording</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Modified Input variable, output, rules, regulations and calculations</td>
</tr>
<tr>
<td><strong>Solution</strong></td>
<td>There is a specific function called “Related Changes” in MediaWiki. By using of this function, we can easily find the historical changes regarding to a specific page.</td>
</tr>
<tr>
<td><strong>Problem</strong></td>
<td>-</td>
</tr>
<tr>
<td><strong>Ideas of refinement</strong></td>
<td>-</td>
</tr>
</tbody>
</table>
5.5 Applicability and generalization

From the test result in section 5.4, the applicability of MediaWiki in this project can be proved. At first, it is important that collecting knowledge inside DA system as well as exploring the approach of structuring the knowledge. Basically, DA system consists of the knowledge from customers (customer requirements, product related knowledge), and also from developer (system developing knowledge). Since DA system integrated knowledge from customers and developers, the main routine of structuring knowledge can follow either the developer’s idea or the system running sequence. From this point of view, we can follow the system running process, and find corresponding knowledge for solving every task in the process. After collecting and structuring knowledge based on whole system process, the processes of electric calculation and geometry sequence need to be considered. Because in design automation system, according to customer requirements and engineering design rules, the calculation have to be done in the system automatically with certain sequence, as well as the geometry should be automatically generated based on macro code with its preset process flow. As a result, we can infer that our method for structuring knowledge that included in the DA system is applicable in any DA system. At this point of view, the approach of setting up the interconnections in both electric sequence and geometry sequence is also reasonable. The specific approach was indicated in section 5.3.

With the implementation of MediaWiki, the most important point is not either how to structure knowledge or model knowledge. The most critical issue that we need to remember is all most all of the knowledge, information or design rationale can be easily presented and these knowledge could be managed by setting of a number of simple interconnections among them. Since the purpose of this project is retrieving, reusing and maintaining the knowledge include in system, MediaWiki performs quite good in such works. Because of its flexibility, easy to use and efficiently managing of knowledge, it can be widely used not only in documentation of DA system, but also other documentation work.
Discussion and Conclusions

6.1 Discussion of solution
To achieve the knowledge structuring objective, the process based approach is used as final method. We took Electric sequence and Geometry sequence separately as the basic structure of the process. This method is suitable for DA system because these two sequences penetrate the whole process either design stage or actual run stage. In the electrical sequence process, process based approach was feasible solution, but in geometry sequence another way of structuring method was used due to the gap between complicated program running sequence and tasks sequence which had been set in the initial stage of developing DA system. The complexity of building up the connections between macro code, macro modules and its functionality was a challengeable task to solve with, and by using of a clear clarification method, the goal was achieved. Furthermore, since the knowledge included in DA system is much more than we had dealt with such as database knowledge and cost estimation knowledge. Such kind of knowledge also should be structured in systematic way in order to achieve a unified documentation approach for a DA system.

6.2 Discussion of method
After deciding to use process based method, we choose Semantic MediaWiki as the platform to present the structured knowledge and manage the knowledge reasonably. This combination achieves most of the requirement we preset while few of the requirement cannot be fulfilled and some of them cannot directly realize. The extra functions from MediaWiki, such as setting of the authorization among a group and searching function which also can search for historical changes brings outstanding benefits to management tasks. The defect of using MediaWiki is that it lacks of some knowledge structuring tools, but it has extraordinary flexibilities on presenting works. The interconnection among knowledge objects is built hierarchically. Some of knowledge representation approaches in MediaWiki need to be improved. For example, we use an indirect way to solve the problem that text inside of picture or image cannot be set as hyperlink, it is still not as convenient as it should be.

6.3 Conclusions and future work
The knowledge and information in design automation system have been structured and management by Semantic MediaWiki. By using these method and application, we achieve the initial goal that finding a reasonable and simple way for structuring knowledge/information and managing documents/knowledge. It is a systematized specification for new engineer. The solution enabled and facilitated system maintenance, updating and support the reuse of functions. The method and approach is applicability and generalization, could be apply to other DA system.

For the future study and work, the following aspects need to be thought about:
• Form theoretical perspective: more specific and systematical method for structuring knowledge approach need to be explored.

• Form a practical using perspective: more user-friendly distribution of knowledge representation on the web need to be developed.

• Discuss the method of structuring and presenting the knowledge regarding the production preparation, cost estimation.

• Exploring the approach of structuring the knowledge included in database is a challengeable task in the future work, as well as production preparation, cost estimation.
7 Reference


