Enabling traceability of design rationale using the concept of product family description (PFD)

Morteza Poorkiany

Master Thesis 2011
(Product Development)
This exam work has been carried out at the School of Engineering in Jönköping in the subject area Product Development. The work is a part of the Master of Science program.

The author takes full responsibility for opinions, conclusions and findings presented.

Examiner: Roland Stolt

Supervisor: Fredrik Elgh

Scope: 30 credits (second cycle)

Date: 2011/10/07

Postal Address: Box 1026
Visiting Address: Gjuterigatan 5
Telephone: 036-10 10 00
551 11 Jönköping
Summary

This thesis work is based on the previous researches in design automation at Sandvik Coromant. The concept of product family description (PFD) has been introduced to the company to improve documentation of knowledge in engineering design process. Current documentation at the company for engineering design covers mostly the design definition part of the knowledge. PFD is constituted by design definition and completed by design rationale. This kind of documentation improves reusing, revising and expanding the knowledge at the company. On the other side, PFD is an input for design programming and a good engineering design description for a product provides more efficiency in design programming.

The project is started by a survey for several principles and applications for knowledge modelling. Product variant master (PVM) and Semantic MediaWiki are selected by the results of the survey. To show the concept of PVM, modelling of a test product is done in product model manager (PMM) software.

The main part of the project is setting up product family description (PFD) by capturing design rationale for the test product, implementing in Semantic MediaWiki. Since the design rationale is not documented, it was recorded during several meetings with the designer of the test product. The description is completed by including the argumentations about the rules, figures, dimensions and etc. Also in the project has been tried to improve and revise the description to make it more simple and efficient. Another objective of the project is to show Semantic MediaWiki as a candidate application for modelling knowledge at the company. In this step the applicability and functionality of both PFD and Semantic MediaWiki is seen.

In the next stage the project findings and company documentation are evaluated. In this step has been tried to show the pros and cons of the project. The emphasis of the evaluation is on PFD and the alternative application.

In the end a conclusion of the whole methods and findings of the project comes with discussion with people who were involved in this work.

Keywords

Product variant master (PVM), product model manager (PMM), product family description (PFD), design definition, design rationale, Semantic MediaWiki
Acknowledgements

This exam work has been carried out during six months at the company, Sandvik Coromant, in Sandviken, Sweden. I have taken effort in this project. However, it would not have been possible without the kind support and help of many individuals and organizations.

I am highly indebted to Johan Hammarlund, design automation manager at Sandvik Coromant, who had kind concern and consideration regarding my living and working requirements.

It is a pleasure to thank Mikael Pettersson, Robert Sikström, Stefan Bergman and Martin Nilsson for their guidance as well as providing necessary information regarding the project and also for their support in completing the project.

I would like to express my gratitude and thanks to my colleagues for giving me such attention and time, especially people at design programming department.

Lastly, words alone cannot express the thanks and appreciate I owe to Fredrik Elgh, head of research computer supported engineering design at Jönköping University, who supported me in any respect from the beginning until the completion of the project.
## Contents

1. **Introduction** ................................................................................................................. 2  
   1.1  Purpose ...................................................................................................................... 2  
   1.2  Delimitations ........................................................................................................... 2  
   1.3  Method and approach .............................................................................................. 3  
   1.4  Outline ..................................................................................................................... 3  

2. **Company presentation** ................................................................................................. 4  
   2.1  Coromill490 product family .................................................................................... 4  
   2.2  Design automation at sandvik coromant .................................................................. 5  
   2.3  Development process at Sandvik Coromant ............................................................ 6  
   2.4  Introducing design descriptions ............................................................................. 7  

3. **Frame of reference** ....................................................................................................... 9  
   3.1  Design automation .................................................................................................... 9  
   3.2  Documentation of product related knowledge ......................................................... 10  
   3.3  General principles for knowledge modelling ......................................................... 11  
   3.3.1  Systems modelling language .................................................................................. 11  
   3.3.2  MOKA .................................................................................................................. 13  
   3.3.3  CommonKADS ..................................................................................................... 14  
   3.3.4  Product variant master ........................................................................................ 15  
   3.3.5  Conclusion ........................................................................................................... 15  
   3.4  Applications for knowledge modelling ..................................................................... 16  
   3.4.1  Semantic MediaWiki ............................................................................................. 16  
   3.4.2  PCPACK ............................................................................................................... 18  
   3.4.3  Design rationale editor ....................................................................................... 19  
   3.4.4  Conclusion ........................................................................................................... 20  

4. **Implementation** ............................................................................................................ 21  
   4.1  Evaluation of principles and applications ................................................................... 21  
   4.2  Modelling PVM by using PMM ............................................................................... 22  
   4.2.1  Model elements definition .................................................................................. 24  
   4.2.2  Conclusion ........................................................................................................... 27  
   4.3  Setting up PFD by using traceability of design rationale in SMW ............................ 28  
   4.3.1  Installation and general functionality .................................................................... 28  
   4.3.2  Implementing PFD ............................................................................................... 29  
   4.3.3  Documenting design definition and design rationale ......................................... 30  
   4.3.4  Improving and expanding PFD ........................................................................... 33  
   4.3.5  Conclusion ........................................................................................................... 38  

5. **Findings and analysis** ................................................................................................. 40  
   5.1  System functionality ................................................................................................ 40  
   5.2  Evaluation the applicability and usefulness of PVM and PFD .................................... 41  
   5.3  Comparison of project findings and company documentation .................................. 42  

6. **Discussion and conclusion** ........................................................................................ 43  
   6.1  Discussion of method ............................................................................................... 43  
   6.2  Discussion of findings .............................................................................................. 44  
   6.3  Systematic evaluation .............................................................................................. 44  
   6.4  Conclusion and recommendations ............................................................................ 48  

7. **References** .................................................................................................................. 50  

8. **Appendix** .................................................................................................................... 51
Introduction

1 Introduction

Sandvik Coromant is a part of Sandvik Tooling and is a world leading producer of tools for turning, milling and drilling. For many years the company has worked with automation of their business processes which has resulted in a stream-lined process for quotation and order preparation, from customer specification to NC programs. This automation is a competitive advantage and enables the company to rapidly deliver custom-engineered product at a low cost. There is continuously a need of new methods and tools to become more agile and efficient in the product development process and in the automation of product family descriptions at the company. Automation of product family descriptions implies the establishment of parametric models and design rules which together constitutes a design space. This digital description enables tailor made products to be generated in a few minutes including all the necessary information for their manufacturing.

Reuse is seen as an important means to speed-up the development process and is partly used today at the company. However, there is a need to further improve the reuse of knowledge and different product descriptions.

1.1 Purpose

To be able to reuse a solution access to the knowledge that once was used for its development is required. Knowledge will be accessible by having a structured documentation. Product Family Description (PFD) has been introduced to Sandvik Coromant by the previous researches to improve documentation. The main objective of the thesis work is to explore the concept of Product Family Description (PFD) by enabling traceability of design rationale and to give suggestions of changes and improvements for further development of the concept.

1.2 Delimitations

The focus of project work is on product family description. The description contains engineering design information. This means handling the knowledge for rules, parts, assemblies and so forth for a product is the main objective of the concept, not information regarding either product development or design programming.
1.3 Method and Approach

In order to achieve the goal of the project, a knowledge modelling system should be defined in the company which is valid for every kind of product families. The project tries to introduce different methodologies and applications for knowledge modelling. PVM as a methodology and Semantic MediaWiki as an application are chosen. PVM is implemented in PMM and Semantic MediaWiki is used for setting up PFD. In order to create PFD, design rationale for the test product by having several meetings with the designers is captured. After implementation, has been tried to improve and revise PFD by making some examples on the test product.

The project has been done in design automation department at Sandvik Coromant with a close collaboration with the designers and tried to work based on the theoretical knowledge of the studied master program especially computer supported systems.

1.4 Outline

The outline of this master thesis is modelling and evaluating product variant master and product family description concepts by working on a test product, improving product family description and coming up with a general modelling system and recommendations for future. Chapter 1 is an introduction of the exam work describing the background, purpose, delimitations, outline and methods of the project. Chapter 2 tries to present the company, explain design automation and different processes and descriptions of product development at the company. Chapter 3 is about a survey regarding different principles and applications used for knowledge modelling systems. Base on the result of the survey implementation of PVM and PFD in PMM and Semantic MediaWiki is shown in chapter 4. The analysis of project findings and company documentation is done in chapter 5. Chapter 6 is the final discussion and conclusion of the thesis work with the recommendations for the future. The references which are used in this project are listed in chapter 7.
2 Company presentation

Sandvik is a high technology engineering group, with advanced products and world leading positions in industry. It was founded in Sandviken, Sweden in 1862 by Göran Fredrik Göransson. At the first, the product range included drill steel for rock drilling and then has developed into a global enterprise in field of materials technology. Now the group has 47000 employees, representation in more than 130 countries [1].

The Sandvik’s business philosophy is to be leader in the selected areas. Products are based on high value content and are developed in close cooperation with market demands.

The group’s goal is to contribute to improving the customer’s productivity and profitability. The products must meet quality, safety, flexibility and economy.

In Sandvik, operations are concentrated into three environments [1]:

- Sandvik Tooling focuses mainly on tools and tooling systems for metalworking applications. Major customers include companies in the automotive and aerospace industries. (Figure 1)
- Sandvik Mining and Construction specializes in rock working equipment and tools used in mining and civil engineering worldwide. (Figure 2)
- Sandvik Materials Technology develops products in stainless steel, special alloys and resistance heating materials as well as process systems. (Figure 3)

Sandvik’s strategy is based on the interaction between a number of strength factors, including advanced research activities, highly value products, own manufacturing and marketing and direct sales to customer.

2.1 Coromill490 product family

In order to make the results of the project clearer, all different steps and processes in this project are done on a test product, Coromill490. Coromill490 is a milling product and used for both face and shoulder milling, especially for applications requiring smaller depth of cut. Coromill490 is available in
several variants in order to improve efficiency and flexibility during working. Figure 4 shows a kind of Coromill490 with 10 teeth. The number of teeth, the direction of rotation and etc could be the parameters that are changed according to marketing demands.

Sandvik Coromant is a global company which has based its business strategies on customized products. The company has a lot of standard products and is important for the company that beside the standard products provides special products based on different customer demands in a short time.

2.2 Design Automation at Sandvik Coromant

Currently there is an increased interest among companies to automate their product processes with the purpose of reducing lead time and human involvement to produce products faster and at lower cost. Sandvik Coromant is a successful company in automation and using computer systems. Automation of different activities started at the company in the late 80’s. The automated activities include: process planning (workflow in production), design with CAD (3D-models and drawings), production preparation with CAM (tool path to CNC machines), steer information to production cells, and measuring preparation (creation of programs to CMM, Coordinating Measuring Machine). The automation of these different activities has resulted in a stream-lined process for quotation and order preparation, from customer specification to NC programs, Figure 5 [2].

For custom engineered products the quotation and order process includes:

- Automatic calculation of price and delivery time based on situation in the production units for each quotation.
- Automatic generation of CAD-models, drawings, NC-data and process planning for each order.
2.3 Development process at Sandvik Coromant

In order to produce a product in Sandvik Coromant, a product project is defined. The project is started by making a project order that describes the scope, characteristics and expected outcome of the project and the conditions of the pre-study. According to the customer and technical specification, project plan and pre-study report are made by project manager and project owner. Next step is the decision to start the project. The project is running in product development and then product introduction steps. The last stage is a final summary of the project which includes outcome, costs and project plan. The summary also describes activities and responsibilities.

All activities from marketing to application programs in order to produce a product can be described in development process in three steps [2]:

1. Product Development: Develop and verify individual instances of a planned product family (includes structural analysis, functional tests, CAD-modelling and building prototypes)
2. Engineering Design: A design space of the product family by rules and associated 3D solid models based on the instances is defined.
3. Design Programming: By using rules as required inputs, the 3D models with information, such as geometries, datum features and named surfaces are prepared to be used when creating programs for product design as well as programs for CAM and CMM.

The output of design programming is 3D models configured for CAM and CMM preparation, quotation drawings, assembly and manufacturing drawings, and customer data.

Figure 6 shows development process with the output design automation system at Sandvik Coromant.
As it mentioned in the objective of the project, there is a lack of a general system at the company for knowledge modelling. The company needs a system to improve the documentation of engineering knowledge for future usage. This study tries to explore different concepts of knowledge management, show the importance of documentation, and provide a system foundation for modelling and management of product knowledge supporting reuse, expansion and maintenance of the system.

Currently people at the company have access to the engineering knowledge within the internal website. First the knowledge is recorded by the designer in word or excel files and then sent to another department to make it accessible for authorized people. By having a look at the knowledge documentation for Coromill490, it is clear that the purpose of documentation has not been mentioned as an important item during developing the product. Low quality of the documentation, difficulty to access to information for design engineers and design programmers, being hard for people to share their developed solutions with the other project members are the problems at the company [2]. The designers might edit the word or excel files during time but updating the shared knowledge on the website is not considered.

### 2.4 Introducing design descriptions

According to the three main company development processes which introduced before, three Design Descriptions are defined to enable documentation and traceability in design automation system, Figure 7 [2]:

![Diagram of Company development process, information and repositories](image-url)
- Product Development, Product Instances Description (PID)
- Engineering Design, Product Family Description (PFD)
- Design Programming, Product Automation Description (PAD)

The reason of introducing these descriptions is to support capturing of design definition and design rationale, facilitate high quality documentation, link the descriptions to supporting documents, models and items and update descriptions easily. The description includes explanation about the product, parts, assemblies and parameters, relation between different parts and rules that describe the design space. This constitute the design definition of the description and by adding supporting information such as purpose and validity of rules, constraints and calculation, the design rationale of the description is made.

The focus of this project is on PFD and trying to improve this concept. A PFD should be able to cover all information about product family architecture, configurations, assemblies, parts (materials, dimensions, properties …) and rules. PFD helps engineering designer and design programmer to maintain, reuse and expand the product knowledge in a design automation system. Traceability is supported by linking statements inside or between descriptions and to knowledge carriers (prototypes, CAD files …).
3 Frame of reference

The important parameter during producing a product is engineering design process and any improvements made to this phase have a considerable effect on both production process and market. The design team tries to identify the information that defines problems and then use their knowledge and skills, along with the available tools, to process that information into a state that defines the solutions. Applying computer technology to the field of engineering design can be a way to improve reusing and prevent wasting time.

This chapter starts by describing design automation, the way of documentation product’s knowledge in an automation system and a survey about several principles and applications used for knowledge modelling system. Based on the results of the survey, one general principle and one application are selecting for further work.

3.1 Design Automation

Computer support can provide good solutions by the automation of tasks that would otherwise be impossible for humans to be carried out. Companies in order to produce customized products need to make use of advanced application systems for automating the work of generating product variants based on different customer specification that is called design automation system [3]. Design automation is a method which uses computer support in order to partly or completely automates the design process [4]. Design automation is based on standardization of parts, systems and procedures in order to have customized products. The establishment of a design automation system is a significant investment in time and money which is expected to give revenues over years. Some achieves by using design automation can be:

1. Efficiency in work
2. Reduction of lead time and costs
3. Better performance and quality of the product
4. Focus of designer on creative tasks

There are some terms such as, Introduction of new products, new variants of existing products, new manufacturing processes, additional or changed design rules due to new insights or changes in standards or legislation, that can affect design automation system and there for, design rules and execution control should be kept updated over time [3]. This goal can be achieved by having a system to manage design knowledge and make it possible to reuse and revise structured knowledge which has been used before.
3.2 Documentation of product related knowledge

Frequent updating of design rules and execution control will become necessity to maintain the usefulness of a design automation system. Significant efforts are required for adapting an established knowledge based system to changes in product technology, new product knowledge, production practices, new customers and so forth [3]. Knowledge documentation should be executed in such a way to make it easy for designers to reuse the old knowledge in a new product family or developing an old solution. This can be done by having a system that can be easily maintained and updated.

Currently at the company, the documentation of engineering knowledge for a product family just shows the results for rules without any description. In this way documentation just can answer to the question “What”. But the definition of a rule is based on the experience of the designer, trial and error, prerequisites, assembling and manufacturing’s constraints and etc. A deeper understanding of rule can be supported by the answers to questions such as Why, When, Scope, Valid ranges of input/output, Origin and so forth [2]. In order to answer to these kinds of questions the concept of Meta Knowledge by using design definition and design rationale is introduced:

Design definition is the object documentation and focuses on the construction and the function of a process (answering question what). Design rationale is the object process documentation and focuses on the argumentation and supporting descriptions for a process (answering questions why, when …) [2]. Meta-Knowledge uses both design definition and design rationale together to constitute the design description. Defining the design description is a way to improve the development process in design automation at the company. Figure 8 shows the concept of Meta-Knowledge.

![Figure 8. Knowledge object and meta-knowledge](image)

Since engineering designers cannot retain all the required knowledge for a design process in their heads, they need to store it in some external sources. Information and knowledge should be captured and stored in such a way to make it easy for reusing or revising. The most effective and desirable way of storing knowledge is an engineering system that is supported by software engineering tools. The knowledge modelling system can be based on general principles such as methodologies or applications such as software.
3.3 General principles for knowledge modelling

A knowledge model can provide structuring and formalization of engineering knowledge to integrate knowledge from various steps, enable concurrent engineering and support engineers in routine design.

This part of the project introduces four general principles as methodologies for modelling product’s knowledge.

3.3.1 Systems Modelling Language (SysML)

Systems engineering is an approach and discipline to deal with complex systems realized through software and hardware solutions [5]. In order to validate requirements or to evaluate the system, System Engineering relies on modelling and simulation methods. Therefore modelling is common practice in System Engineering to deliver functional specifications, data flow description, or system structure definitions through the use of modelling techniques.

SysML is a general purpose modelling language for systems engineering applications. It supports the specification, analysis, design, verification and validation of a broad range of systems and system of systems. These systems can include hardware, software, information, processes and facilities [6].

SysML can represent systems, components, and other entities as follows [7]:

- Structural composition, interconnection, and classification
- Function-based, message-based, and state-based behaviour
- Constraints on the physical and performance properties
- Allocations between behaviour, structures and constraints (e.g., function allocated to components)
- Requirements and their relationships to other requirements, design elements, and test cases

SysML is designed to provide simple but powerful constructs for modelling a wide range of systems engineering problems. In particular, the language provides graphical representations with a semantic foundation for modelling system requirements, behaviour, structure and parametrics, which is used to integrate with other engineering analysis models.

In order to represent hardware, software, information, personnel, procedure and facilities, SysML uses the concept of the block as the basic unit structure. A block diagram describes the system hierarchy and system classifications. Since SysML deals with the realization of the system hardware and software blocks, knowledge is captured through models and stored in a single repository. This can be a possibility for heterogeneous teams to generate specifications and thereby increase communication through all teams.
SysML includes nine diagrams to make it a small language and easy to learn and apply. The focus of this survey is on requirement diagrams and parametric diagrams to know more about how to implement constraints and rationales in SysML.

Requirement Diagram: A requirement specifies a capability or a condition that should be satisfied. Requirements are used to establish a contract between the customer and those responsible for designing and implementing the system.

SysML Requirement Diagrams provide modelling constructs to represent text-based requirements and relate them to other modelling elements. These requirement modelling constructs are intended to provide a bridge between traditional requirement management and other SysML models. Requirement diagrams show requirements, rationales, test cases and relationships. Requirements can also be shown on other diagrams to illustrate their relationships to other modelling elements.

Parametric Diagram: Identifies constraints on system property values such as reliability, performance, mass properties and etc. It provides the ability to integrate the specification and design models with engineering analysis models. This diagram uses model elements called constraint blocks that define generic or basic mathematical formulas. A constraint block typically contains one or more constraint parameters, which are bound to properties of other blocks in a surrounding context where the constraint is used. All properties of constraint block are constraint parameters.

Figure 9 shows an example for parametric diagram with six constraint properties (in the middle) which are linked together. Systems parameters are split between input (green) and output (blue).

![Figure 9. An example for parametric diagram in SysML. [5]](image-url)
3.3.2 Methodology and tools Oriented to Knowledge based engineering Applications (MOKA)

Having a way of collecting and structuring the engineering knowledge associated with designs can make it easy to plan and organize the process of building knowledge based engineering applications.

Knowledge Based Engineering (KBE) concerns the computerization of processes associated with industrial products, usually routine design [8]. KBE is a computer system that enables the creation of a fully engineered, best practice design by storing the experience, geometry and data that are related to a product.

The development of a methodology to support the deployment of knowledge based engineering applications is the subject of MOKA [9]. Providing a way of developing and maintaining KBE applications and reducing costs, risk and lead time are the main purposes of MOKA.

In brief can be said the methodology of MOKA follows different steps as:

- Determine type of KBE system to have a conceptual specification of the KBE application
- Create a project plan by considering different issues
- Structure the raw knowledge in a model according to rules and constraints, use MOKA Modelling Language
- Implementation of model in to a KBE system

The result of the project will be a formalized methodology to analyze and model products, design processes and associated knowledge.

MOKA defines different meta classes to structure the product model. Also different meta views are pre-defined for engineering knowledge modelling such as structure, function and behaviour, as well as the relation between them. In figure 10 a structure view for MOKA is shown.

![Figure 10. The default meta-classes for the MMI view structure in MOKA [8]](image-url)
3.3.3 CommonKADS

Knowledge has to be organized and structured in order to be recognized and handled in the company. Therefore managing knowledge has become an enterprise task in the company.

CommonKADS is a methodology to document knowledge engineering and management. It acts as a baseline for system development and research projects. Furthermore the CommonKADS methods are nowadays in use for purposes other than system development, such as knowledge management, requirements capture and business process analysis. CommonKADS originates from need to build industry-quality knowledge systems on a large scale, in a structured, controllable and repeatable way [10]. CommonKADS is the leading methodology to support structured knowledge engineering and gives tools to corporate knowledge management, also it supports the development of knowledge systems that support selected parts of the business process.

CommonKADS is applicable for both software engineer and knowledge manager [11]:

- For software engineers who are in the business of building knowledge-intensive IT systems that need to satisfy the organization needs of customer, CommonKADS offers a de facto standard for system development and ensures a high – quality solution based on reusable components and supported through practical guidelines and tools.
- For knowledge managers, CommonKADS offers methods to create coarse grained description of knowledge-intensive tasks within the overall business process as well as techniques for detailed knowledge analysis, knowledge development and knowledge storage.

CommonKADS modelling consists of a number of predefined stages in a predefined sequence in order to find out about the customer requirements, specify and design the system, program, test and deliver it. CommonKADS has a predefined set of models, each of them focusing on a limited aspect, but together providing a comprehensive view. This view is shown in figure 11.

![CommonKADS model suite](image)
3.3.4 Product Variant Master (PVM)

In industrial companies where many people are involved in developing, marketing, selling, producing and servicing products, specifications make up an important part of daily life. A specification can be defined as a description which can transfer needs or intentions from one group of people to another.

Specification processes are a part of company’s operational system and denote the business processes which analyze the customer’s needs, create a product which is adapted to the individual customer, and specify the activities which have to be performed in connection with, for example, purchasing, production, assembly, delivery and servicing of the product concerned [12]. These activities can be supported by building up product configuration systems. A configuration system contains a description of the company’s product range and rules for how a customer-specific product variant can be designed.

Product Variant Master is an operational tool to model and visualize a product range. In general, a family in PVM can be known as [12]:

- Part-of structure which shows the components included in the product
- Kind-of structure that shows the variants available

For modelling PVM three different views exist such as, customer view, engineer view and production (part) view [12] that are related to one another. A very important aspect of a product variant master is the relationships between these views. In order to set up a PVM, first of all the customer view should be mentioned according to customer demands. Once a version of customer view was finished, designer starts to create the engineering view by considering important rules and constraints. After that the version of production view which can be the list of parts is created.

A general formalism for a product variant master is based on: Classes, attributes and rules. Class is a group of objects with the same structure or function. Each class has a unique name and can contain attributes such as price, weight, code number and etc. Attribute describes the class and is formally declared with a name followed by a value interval and a measurement unit. Constraints describe rules for how classes and attributes can be combined. Constraints described in text are placed under the list of attributes.

3.3.5 Conclusion

Organizing and managing engineering knowledge should be executed in such a way to be applicable for a wide range of people, manager, analyzer, modeller, designer and etc. by having a look at the four mentioned principles, SysML seems as a simple way to show the rationale, requirements, constraints and rules by using the concept of the block diagrams, while CommonKADS looks more
like a dominant method to manage the knowledge. In CommonKADS, all the information from
design to delivery is shown in a simple way. Storing the experience, geometry and data that are
related to a product and show them within different classes and views is an outstanding for MOKA.
When it comes to reducing costs, risks and lead time in a project, providing a way of developing and
maintaining KBE makes MOKA more specific. Product variant master gives a general overview of
the product according to sub or super parts with the relations between different components which
all the model can be seen in a big piece of paper.

3.4 Applications for knowledge modelling

One goal of the project is to use an appropriate application for system modelling which could be
used for recording, sharing, re-using and managing information. In following, three applications are
introduced which are used for knowledge modelling of product families. In the end one application
will be selected for modelling PFD. Selecting application should be according to the project objective
and company situations.

3.4.1 Semantic MediaWiki (SMW)

In order to know Semantic MediaWiki, is better to start by introducing Wiki and MediaWiki [13]:

A Wiki is a website that allows the creation and editing of any number of interlinked web pages via a
web browser using a simplified mark-up language. Wikis are typically powered by wiki software and
are often used to create collaborative works. Examples include community websites, corporate
intranets, knowledge management systems, and note services. Wikipedia is the best known example
for a wiki as a free encyclopaedia that everyone can edit.

MediaWiki is free server based software and designed to be run on a large server farm for a website
that gets millions of hits per day. When a user submits an edit to a page, MediaWiki writes it to the
database, but without deleting the previous version of the page, in order to make it easy for reverting.
MediaWiki also can manage image and multimedia files, which are stored in the file system.

Semantic MediaWiki [14] is a free open-source extension to MediaWiki that lets to store and query
data within the wiki’s pages. Semantic MediaWiki is also a full-fledged framework, in conjunction
with many extensions that can turn a wiki into a powerful and flexible collaborative database. All
data created within SMW can easily be published via the semantic web, allowing other systems to use
this data seamlessly.

The integration between MediaWiki and SMW is based on MediaWiki’s extension mechanism:
SMW registers for certain events or requests, and MediaWiki calls SMW functions when needed. SMW does not overwrite any part of MediaWiki, and can be added to existing wikis without much migration cost.

Semantic MediaWiki helps to search, organize, tag, browse, evaluate, and share the wiki’s content. Many reading features of the wiki should be obvious to users. For example, numerical values using units of measurement will display tooltips with unit conversion. While traditional wikis contain only text which computers can neither understand nor evaluate, SMW adds semantic annotations that allow a wiki to function as a collaborative database. While the first appears to make things more complex, it can also greatly simplify the structure of the wiki, help users to find more information in less time, and improve the overall quality and consistency of the wiki.

The purpose of Semantic MediaWiki is to allow users to improve the structure and organization of the knowledge in a wiki by adding simple, machine-processable information to wiki articles. With this additional information, we can greatly improve searching, browsing, and sharing the wiki’s knowledge; both within the wiki’s pages and from external computer programs.

Editing semantic data in SMW is to simplify common tasks for users of the wiki. browse wiki, export pages to RDF, search by property, and semantic search are the applicable options of SMW that some of them are shown in figures 12 and 13:

![Figure 12. Semantic search](image)

![Figure 13. Property search](image)
3.4.2 PCPACK

PCPACK is a powerful business solution that supports the retention, sharing, management and re-use of knowledge to the organization. Since it is difficult to capture and organize engineering knowledge from different involved people, PCPACK provides tools that maximize the quality and quantity of the knowledge acquired whilst minimizing the time and effort required from experts valuable to the organization [15].

PCPACK is a user friendly and versatile application. It provides a publishing function so that the knowledge base contents can be automatically displayed as a web site or as more traditional documentation. This knowledge will be available for people who need to have access to it.

PCPACK uses a number of tools that provide user-friendly graphical interfaces to structure knowledge. For example categories, sub-assemblies and sub-components can be shown in PCPACK. Also defining and presenting relationships and properties of pieces of knowledge is another applicability of this software.

In order to make sure that knowledge shared around the organization is complete, correct and relevant, the knowledge must be validated with experts, which in PCPACK the expert is presented with knowledge that is quick and easy to check for errors and omissions.

As mentioned above, PCPACK is a powerful tool and can share knowledge in web formats for people who need. In PCPACK, websites can be created quickly by translating the contents of a knowledge base into the appropriate web formats. The visual style and functionality of websites created by PCPACK can be fully customized using templates. Once a template has been created, it can be used repeatedly through the organization when publishing web sites with PCPACK.

In order to re-use knowledge, PCPACK uses a technology called XML, which is fully compatible with modern web technologies such as the semantic web that provides a formal, machine-readable content that can be automatically translated into other formats and used for the automatic creation of software programs.

In PCPACK, ten tools are defined to make the knowledge modelling more easy and flexible; five acquisition and modelling tools and five specialized tools.

Acquisition and modelling tools:

- The Protocol Tool is used to analyze a transcript or other text documents
- The Ladder Tool is used to construct hierarchical diagrams
- The Diagram Tool is used to construct network-style diagrams
- The Matrix Tool is used to construct two styles of matrix
- The Annotation Tool is used to create web-style pages for each knowledge object
Specialized Tools:

- The Admin Tool is used to access and manage knowledge bases.
- The Publisher Tools used to create websites
- The Diagram Template Tool is used to create templates used in the Diagram Tool
- The Equation Editor is used to create equations for use in the Annotation Tool
- The Tool Launcher is a wizard tool allowing easy access to the other tools

The modelling tools are shown as examples in figure 14:

![Modelling tools](image1.png)

Figure 14. From left to right: Protocol, Ladder, Diagram, Matrix, and Annotation Tools [15]

### 3.4.3 Design Rationale Editor (DRed)

Engineering design is based on the use of past experiences. It is known that engineering designers frequently need to revise previous design solutions and understand the rationale for their generation.

Design Rationale Editor allows designers to record their design rationale at the time of its generation and deliberation. The design rationale is displayed in a document as a graph of nodes linked with directed arcs. The user creates the nodes by choosing from a predefined set of element types [16].

DRed improves the richness and clarity of the recorded information and also has a beneficial effect on the design behaviour by prompting design thinking and helping designers to view their design process. It assists with managing design tasks, captures design rationales as they are created without an additional overhead, and reduces the need for written reports.

The functionality and interface of DRed can be described by four main applications as: [17]
• Diagnosing a problem (problem understanding): DRed starts by forming theories about potential causes of the problem, and looking for evidence to support or refute them.

• Designing a solution (solution synthesis): In this step a solution to the problem is designed.

• Completing a standard checklist template: A part of the design procedure of the company is for the final design definition to be checked against a checklist of standard criteria, or associated effects. This can be done in DRed by using a standard checklist template.

• Communicating the final design and its rationale: The final step is to prepare a narrative description of the solution, complete with details such as the rework procedures required to apply the solution to engineers in service. These descriptions are best written as standard word-processed reports.

Figure 15 shows an example for a DRed document. All the elements in a DRed chart are given a “traffic light” status. When an Issue is first created, its status is open and its colour is accordingly amber. Decisions are captured by manual changes of status, reflected in changes of colour to either green or red. Once the designer is satisfied that the diagnosis is complete and correct, he or she will record this by changing the status of the top-level Issue to Resolved, shown by the question mark turning green. Conversely, if an Issue has to be marked Insoluble, it is turned red. An important consideration in DRed is that there should be no hidden information and everything is displayed directly on the chart.

![Figure 15. An example for DRed document [16]](image)

### 3.4.4 Conclusion

Improving data structure by using categories and searching specific information according to user’s queries are the advantages of SMW. Editing the contents of a page can be done in the easiest way in SMW. On the other side, PCPACK is an integrated suite of 10 knowledge tools designed to support the acquisition and use of knowledge. Analyzing knowledge from text documents and structuring knowledge using various knowledge models makes PCPACK a powerful system. DRed is a simple and unobtrusive software tool that allows engineering designers to record their rationale as the design proceeds. It allows the issues addresses, options considered, plus associated pro and con arguments (arguments for or against an answer), to be captured in the form of a directed graph of dependencies.
4 Implementation

Currently there is a lack of a knowledge modelling system at Sandvik Coromant. The modelling system can manage the way of capturing knowledge and information and making documentation. Having such a system improves reusing of recorded information. The system should fulfil some requirements such as [18]:

- The information must be stored in a structured way on a system-independent model
- It has to be possible to view where the information currently is used and has been used
- The information has to be quality assured
- The information has to have an owner
- The use of the information must be easily understood
- The information must be accessible easily
- It must be easy to update
- The information must be possible to versioning

The discussed principles and applications in chapter 3 can be used partly or completely as desired modelling systems. The project continues by choosing the most suitable principle and application, as alternatives for the case company to document the engineering knowledge.

4.1 Evaluation of principles and applications

Sandvik Coromant has a wide variety of specific products and needs a simple and user friendly application for documentation considering reuse, maintain and expand the knowledge. The best way to make simplification in a product is to have a general view of that product with all relations and constraints. This can be done by the concept of product variant master. The concept of product variant master can be implemented easily in product model manager software.

On the other hand company needs a professional search engine within documentation to make the information accessible which in a good way exists in Semantic MediaWiki. Semantic MediaWiki by using different searches that mentioned before can be a good applicable tool according to company’s situations.

Briefly, two software, Product Model Manager and Semantic MediaWiki are selected for modelling PVM and PFD for the test product, Cormill490.
4.2 Modelling Product Variant Master (PVM) by using Product Model Manager (PMM)

Product Model Manager is user-friendly software which is provided by Incore Systems to support documentation of configurators, bill of materials and other kinds of product data.

Product model Manager is a good tool for documentation of complex product information. It includes three interrelated modelling environments such as Product Variant Master, Class Information Cards, and Product Explorer. In Product Model Manager, product structures can be built fast and easy by using drag and drop of many different types of elements.

By using PMM a wide range of benefits can be achieved by the company such as [19]:

- Improved internal communication
- Fewer errors in product specifications
- Less resources needed for documentation of product knowledge
- Ensuring that knowledge stays in the organization
- Reduced lead time for the creation of product specifications
- Reduced costs for product specifications
- Can be used with minimal or no training
- Provides standard interface for data exchange

Three applications are available in PMM [19]:

- Bill of Materials: Product Model Manager provides an extremely user-friendly modelling environment, in which bills of materials can be created as a basis for manufacturing and for documentation.
- Product Data Management: Product Model Manager is a user-friendly alternative to existing PDM systems found on the market today. Product Model Manager supports the management and structuring of product information in a simple way. Furthermore, it provides a flexible basis for product data management through the Class Information Cards. Class Information Cards can easily be adapted to include exactly the fields which a particular company needs.
- Product Configurators: Product Model Manager supports the development and maintenance of product configurators. It allows users without IT-knowledge or modelling prerequisites to participate in the development of a product configurator, by providing user-friendly modelling environments.
Modelling:

The goal of this chapter is to explore Product Variant Master modelling by using Product Model Manager software. To make it clearer, modelling is done with the test case, Coromill490.

The software is available for two months free and can be downloaded on the IncoreSystems website.

In figure 16, a screenshot of the software which includes Product Variant Master, Class Information Cards, Product Model Explorer, and Model Elements environment is shown.

Three kinds of classes are included in Product Model Manager: Root Classes, Part of Classes, and Kind–of Classes. The creation of a new model starts with the insertion of a Root Class. Classes can be inserted by using drag-and-drop and via the context menu. To drag-and-drop, one should click on a desired class in Model Elements window, drag it to the Product Variant Master sheet and drop it. Also by right clicking on the Product Variant Master sheet, the context menu will be appeared for inserting a class.

![Figure 16. The environment of PMM](image-url)

Figure 17 shows clearly the concept of product variant master, different model elements and relations between the classes.
The left side of figure 17 describes the product’s generic structure, such as Front Part, Tool Body and etc. The right side shows the variants available. There are five kinds for Coromill490. Similarly, the figure shows that the Tool Body is available in three variants. In general, the left side (known as the part of structure) in a product variant master shows the components included in the products, while the right side (known as the kind of structure) shows the variants available [12].

4.2.1 Model Elements Definition

A class is defined in accordance with the object-oriented paradigm as a group of objects with the same structure or function. An example of a class is Front Part, which can consist of subclasses such as; Tool Body. Each variant of the Tool Body has descriptive attribute. Attribute describes the class and can be weight, price, code number, and etc. Attributes are inserted in the same way as classes.

Classes take part in two hierarchies, which are known as part-of and kind-of structures. A class can contain one or more classes in a part-of structure. If it is read upwards, the class on the higher level is denoted a super-part. If it is read downwards, the class is denoted a sub-part [12]. In figure 17, the class, “Coromill490” is a super-part with respect to the class “Front Part”. The class “Tool Body” is a sub-part with respect to the class “Front Part”. Each class has a unique name in order to avoid misunderstandings in the interpretation of a product variant master. For practical reasons, the class name should be relatively short. When there is a need for more explanation, a description is used.

In Product Model Manager, it is possible to add engineering rules to a class. There are two ways to show the rules; Symbolic rule and Table rule. Rules are added to the classes in the same way as attributes. Symbolic rule is used when there are different values in an attribute related to different model elements and can move up or down in the model hierarchy to insert the desired value or element. Figure 18 shows an example of a table rule in PMM.

Tables from MS Office programs, i.e. Word and Excel, can be pasted in the Product Model Manager rule editor, and tables from Product Model Manager can be pasted into MS Office programs. In a table rule, the possibilities for inserting class, attributes and values are the same as in the Symbolic rule type.
Adding comment is possible in Product Model Manager in order to write a text. Picture files can be shown in a model as well. Many picture formats are supported in Product Model Manger. Also there is an applicability to add an external file or document, such as Excel, Word, PowerPoint, etc as supported documents.

**Class Information Cards:**

As default, the Class Information Cards consist of the following folders (can be changed in the MasterDataConfiguration.xml file):

- Basic Information
- Relationships
- Product Knowledge
- Attributes and values
- Rules
- Comments
- Pictures
- Implementation Details

The Basic Information folder allows defining information such as Card responsible, Product responsible, Class description, etc.

The Relationships folder shows which other classes, the class has relations to and provides hyperlinks to these. This means that navigation between Class Information Cards is possible.

In the folder Product Knowledge, attributes, rules, comments, and pictures can be managed.

It is possible to customize all the folders, fields, and lists according to individual preferences in the Class Information Cards.

In figure 19 a Class Information Cards is shown.

**Product Explorer:**
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. The Product Explorer view is shown in figure 16.

In the “Edit” menu the “find” function can be activated to search for a class, attribute or value by the name.

Class Information Cards can also be seen in html format by using the function “report”. This allows searching for specific elements and importing the Class Information Cards to e.g. MS Word.

For modelling Coromill490, the engineering knowledge that is documented in Excel files and is available on the Sandvik’s data base has been used. Since there is no any special way to organize, edit, and modify the knowledge, some parts of the knowledge has not been updated and was needed to discuss with the product designers during the modelling. The modelling started by adding part of classes and kind of classes for different Coromill490 variants. Every comments, rules, and attributes
are inserted according to the description above. In some cases has been preferred to keep some documents as Excel files and just add them to the model as external references. The modelling is done for the whole Coromill490 product family and revised continuously by discussing with the designers.

A manual introduction is available on the Incore System’s website which has been used for training.

Since the number of variants in Cormill490 are high and is not possible to show the whole modelling in one screen shot, in figure 20 has been tried to show a part of Product Variant Master for Coromill490 which consists comments, rules, and pictures.

![Figure 20. Modelling Coromill490](image)

### 4.2.2 Conclusion

Product Variant Master is an operational tool for modelling and visualizing a product range. In simplified terms, it can be said that a product variant master contains a description of the product range and the associated knowledge all described on one large piece of paper. Product Variant Master is a way to structure the product model and the configuration system. It makes the system easy to maintain and offer an optimal division of work between the configuration system and users. Product Variant Master is an applicable tool to analyze the product range with a general overview especially when there is large number of variants.
Product model manager is user friendly software for implementation and there is no need of training for a new user. Regarding the product family information related to components, parts or variants, small tables, rules, pictures and comments can be stored beside those components or parts in PMM. But when there is a need of more tables, pictures, rules, description, argumentation and etc, to provide design rationale it is better to be saved in external files such as Microsoft Office. These files can be linked to a component or part as supported documents. By clicking on the link the file will be shown. Searching in PMM is available for class, attribute or a value within a product family. It is not possible to search within different families in PMM.

4.3 Setting up PFD by using traceability of design rationale in Semantic MediaWiki

As it mentioned before, a PFD constitutes with both design definition and design rationale. A description should contain the information about product, assembly, parts, features, functional structure, parameters and rules. Theses all constitute the design definition. The description is completed by adding the design rationale, such as calculation, analyses, field test, underlying principles for design, assumptions, constraints, context, valid ranges of parameters and validity of rules [2]. Linking design rationale to design definition in a PFD is called traceability of knowledge.

4.3.1 Installation and general functionality

Since Semantic MediaWiki is an extension for MediaWiki, first MediaWiki should be downloaded and installed. MediaWiki is free software and can be downloaded from its website. A web server, PHP (programming language) and a data base server are pre required to install the software. These all with a user manual describing the installation are available on the website and can be downloaded. After installing MediaWiki, Semantic Media Wiki can be downloaded and installed according to the available description on the website. The description also describes the software in details and the way of implementation.

MediaWiki uses the concept of wiki pages to record the information. There are several ways to start a new page. One way is by writing the name of the page in search window and following to the created page, typing the text and in the end saving the page. Another way is by creating a link to an article that doesn't exist, a red colour link will be displayed. By clicking on the red link the page will be appeared that the text can be typed and saved. The saved page can be edited by pressing the edit tab and making changes. Also a preview of page can be seen before saving, helping the user to check the text before saving. Every change in a page can be seen by using show changes button. Deleting and renaming the page can be done by using the tabs, by the way all the versions of a page is kept in the memory of the software even if it is deleted. Most kinds of files and images can be used in
MediaWiki by uploading them. There is a special way to show the uploaded file in wiki by using File syntax. For example the syntax “[[File:imagename.jpg]]” can be used to show an image. It is the same to show other files by writing the name and the format of that file. Text can be typed in italic or bold fonts by using "text" or "text" symbols. Making a table is totally different by Microsoft office. MediaWiki uses mark up summary to create a table. Each content of the table should be written in a new line. Figure 21 shows how making a table is in MediaWiki.

<table>
<thead>
<tr>
<th>You type</th>
<th>You get</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orange Apple</td>
<td>Bread Pie</td>
</tr>
<tr>
<td>Bread Pie</td>
<td>Butter Ice cream</td>
</tr>
</tbody>
</table>

Figure 21. Making table in MediaWiki

Figure 22 shows a screenshot of the main page in Semantic MediaWiki which is installed and used to set up PFD for Coromill490. The tabs on the top of the page are the same for every page but can be managed by the admin of the software. In discussion, users can make comments and discuss regarding an issue or item. As it explained before, the page can be edited and deleted by using the tabs. In the tab “move”, the name of the page is changed and the privacy of the pages can be managed in the tab “protect”. The page would be selected as “watch” or “unwatched” to make it easier to pick out in the list of recent changes. In the navigation area, the admin can provide some additional information regarding community portal, events or a help page. Below the navigation area, is the search window. Button “Go” is used to go to a page with the exact name if exists. Button “Search”, searches for the pages contain the text. Below the search window is the toolbox area which can give information regarding links, changes, versions and etc about a page. The link of special page can be used for maintaining the software, finding the list of pages with categories, properties and etc, managing users and especially having access to the semantic toolbox. The important parameters will be explained in the follow.

When installation of Semantic Media Wiki is finished, is time to record both design definition and design rationale in the software.

4.3.2 Implementing PFD

During setting up the PFD, has been tried to use the concept of part of classes as in PVM. The product family is explained according to the classes (the articles in figure 22). For example in figure 22, the PFD of Coromill490 is described by linking to seven articles. These articles are the same as main classes in PVM. In this case by having look at the main page of the PFD the general view of
Implementation

the product’s structure is shown up. This view represents the most important parameters involved in the PFD. In the PFD of Coromill490 for each article of the knowledge, a wiki page is created and named to that article. The documentation of the article is placed in that page. The articles explain different parts of the product. An article can contain supporting documents. Supporting documents such as; excel, word and etc can be added to a page for more information by uploading the file on the software and then link it to that page.

![Figure 22. SMW, main page](image)

### 4.3.3 Documenting design definition and design rationale

Current documentation at the company for product families contains just design definition. In order to set up a PFD, design rationale should be recorded as well.

Setting up the PFD for Coromill490 is started by capturing design rationale. During several meetings and discussions with the designers, the rationale behind every rules and knowledge discussed and documented. First, the design rationale was recorded in word and excel files, modifying and revising continuously. When capturing design rationale is finished, is time to store the knowledge in Semantic MediaWiki.

An example of documenting design definition and design rationale is shown in figure 23 for a component. One page is created to contain the required information for the component. Every knowledge regarding that component such as; pictures, tables, rules and etc is stored in this page. This page contains both design definition and design rationale of the knowledge and includes a
complete description. The text describes different parameters that are involved in, to design the component. It also includes the principle for designing, the validity of the component in the product, the rules and their validity for Coromill490 variants. This part can be the design rationale of the description (answering question why? and ...). Then defined dimensions are used in the rules. The rules are shown in the table. This is the design definition of the description (answering question what?).

![Figure 23. Design rationale and design definition for a component](image)

Another example of PFD for a component is shown in figure 24 with a text and a picture describing the rules. All information needed for the component such as; pictures, tables and rules should be captured in this page. The text describes the rationale behind designing. The picture gives a better understanding of the dimensions. In the end, the definition part of design shows the valid results for different kinds of the component. If there is a need of adding supported documents to a page, the file can be uploaded to the software and be linked in to the page by writing [[Media:Filename.xlsx]] in the page. By clicking on the link the file will be shown. An example for an excel file is shown in figure 24.

![Figure 24. Design rationale and design definition for a component](image)

All the articles shown in figure 22 are explained in the same way as the mentioned examples in order to make the PFD. The pages contain design definition and completed by design rationale.
When the engineering design knowledge is captured, it should be accessible for reusing. This can be done in Semantic Media Wiki by using a good search engine. In the previous chapter, different semantic searches were introduced and here is trying to show them by some examples. Two main ways for a semantic search are by using properties and categories. Categories are used as universal tags for articles, describing that the article belongs to a certain group of articles [14]. They are a means to classify articles according to certain criteria. Categories are used for structuring information and then searching for the information based on that structure. Property is like key word information belongs to an article with a value that makes it specific.

An example for category can be seen in coupling documentation. Figure 25 shows the wiki page for coupling. It begins with a text describing the rationale and then links to two pages for two variants. Coupling 1 links to five and Coupling 2 links to seven pages which contain information for different kinds of coupling. By adding a short description as “[[Category:Coupling and Connection]]” to all these pages, all of them can be added as a category, coupling and connection.

![Figure 25. Making a category of coupling and connection for Coromill490](image)

When a category is created it can be searched in semantic search. In the query window the added text is written which asks for all the pages that are tagged as this category. Figure 26 (left side) shows the semantic search and results for the category coupling and connection.

This search can be more specific by using properties. The principle of property is to look for a document that includes a very special kind of information. To show an example for property, a UDF is considering. The UDF is common for two variants of Coromill490.
Here is trying to show how a mentioned page that includes information about the desired UDF can be found among a large number of pages. In the page which contains the documentation for this kind of UDF, a short description as “Property:[[UDF::value]]” for both Coromill490 variants can create the property UDF, with a value. By writing “UDF” in additional data window (figure 26 right side), the two pages that contain the UDF’s information are shown as results.

In the special page of Semantic MediaWiki, a number of options are available which help the user to manage the information. The most important options are:

- List of all pages in the wiki
- List of all redirects articles
- List of all categories
- List of all properties
- Recent changes in Semantic MediaWiki
- List of all uploaded files
- View deleted pages

According to these options, users can get the information regarding the files, categories and etc, directly on the software in order to reuse, revise or expand the documentation.

### 4.3.4 Improving and expanding PFD

In a PFD an important item is to store the knowledge in the best place and prevent recording the same knowledge in different places. For example in documentation of Coromill490, there are some
information, tables or values that are general for a range of parts and have been stored for each of those parts separately. In order to prevent duplication, documentation can be done in two categories; one as a general category, contained general information which is valid for a range of parts, the second one is a specific category for the knowledge which is valid just for the specific part. The directory for general and specific knowledge can be those wiki pages that contain either general or specific sector of the knowledge for the parts. This solution can be clearer by showing an example. Figure 27 shows a design rule table of a component containing the dimensions, limitations and parts. This table is used with partly the same values for two Coromill490 variants. The differences in the tables are the values for L1 and DDC which are associating specifically for the parts. The table can be divided into two categories such as specific category and general category. The information for L1 and DDC can be captured in the specific wiki pages of the component for two Coromill490 variants (specific category), figure 28. The rest of the table can be stored in the page which is valid for both variants (general category), figure 29. Also L3 column is eliminating since as a designer point of view it is clear that L3 is calculated by the rules.

A PFD should contain all required information of parts and assemblies for the engineering design. Some parts and components are common in different product families or are the most important
Implementation

and used regularly in a product family. Sometimes the designers need to have a quick access to the whole knowledge of these parts. In this case designers prefer to have a structured way of documentation for such parts to make it easier for them when are looking for special information. Having a pre-defined description or template makes it easier for designers in both recording and searching information for those special components or parts. The template can be varied structurally depending on the object but includes a complete description of the part such as: title, introduction and details, responsibilities, collaborates, pictures and so on. According to the object, the template should be pre-defined and a short guideline can describe the template’s structure for future works. In order to show how to define and create a template, again the UDF in Coromill490 is considered. The UDF description shows different sections, their sequences and necessary information that should be stored:

- **Title of the UDF**
- **Introduction**: A short introduction of what kind of UDF it is and what purpose it has.
- **Dependencies**: Another UDF might be needed before this UDF can be mated into the part. If so, it should be stated here.
- **Details**: Detailed description of the UDF. Some examples could be:
  - Manufacturing requirements or limitations in the production may be taken into consideration
  - Built in parameters
  - Description of dimensions and parameters
  - Rules
  - Tables describing certain parameters for certain appearances
  - Descriptions preferably with pictures showing dimensions and entities used in the UDF
- **Selection**: It has to be established when the UDF shall be used. A table showing the different selection alternatives is recommended to use if there are several types of UDF depending on different circumstances such as insert type, right hand, left hand and so on.
- **Surfaces**: The surfaces of the UDF often require names on each face as NC-preparation. Solid Body name could be displayed as well.
- **References**: Description of the references used for mating the UDF in a part. The references may be completed with:
  - A picture showing where they are located
  - Information about layers
  - What is the reference for this UDF (may be a specific UDF)
- **Expressions**: Some of the parameters in the UDF require to be sized in order to get the right appearance to its purpose. The expressions would be displayed with a description of each one. If limits for the expressions are established, they should be displayed here.
- **Output features**: Important features may be described here with a picture showing the location and current layer for each feature.

This UDF description has been declared by the purpose of displaying the required information mostly in design programming but also to make the UDF available for other users. The template shows the way of recording the knowledge for any kind of UDF.

This way of documentation has been implemented in Semantic MediaWiki to show the applicability of using the templates. An example of template for UDF is shown in figure 30. A wiki page is created for the template. According to the description different sections are defined. The template can be more easy and clear by using pictures.
Implementation

Figure 30. The template for UDF
4.3.5 Conclusion

The main objective of this thesis work is exploring the knowledge modelling system based on PFD, implementing in Semantic MediaWiki. PFD by considering both design definition and design rationale is a good alternative for the company to record all parts of knowledge required for engineering design in a product family. The PFD could be more completed and detailed if it was created in parallel work during designing the product. On the other side, Semantic MediaWiki by creating wiki pages makes it applicable for storing and classifying a wide range of information such as; text, image, table and etc.

In order to show explicitly the facilities and pros and cons of the system modelling, a short discussion regarding different issues of implementation is adding:

**PVM structure and its usage in the modelling system**: Since in Semantic MediaWiki using tree diagram is not possible, PVM cannot be modelled in Semantic MediaWiki as the same as in PMM. But modelling the part of structure for a product family is possible to describe PFD based on the parts of the product.

**Searching facilities**: Searching in Semantic MediaWiki is according to text and title or semantic search by using categories and properties. Considering semantic search items during making documentation is necessary. The result of semantic search especially when properties are used depends on the view of the person who has introduced the properties in documentation. Semantic MediaWiki provides a list of all categories and properties which are used in pages. The list can help the users to know about or edit the search items. In semantic search the name of property or category should be written correctly to get the right answer. In this project modelling was done just for one product family. If more than one product family are used, searching cross product families will be possible in Semantic MediaWiki as well.

**Linking supporting documents**: Text, tables and pictures can be added to a page. When there is a need of adding external documents, every kinds of files such as; word, excel and etc can be added to the page. First the file should be uploaded to the software and then a link be created in the page with the same name as the file. By pressing on the link the file will be shown.

**Version control**: Every page can be edited in Semantic MediaWiki. Information can be added or deleted in a page easily by just pressing the edit tab. All the activities or changes in a page are shown in the “Recent Changes”. There is a facility to upload a new version of an existing file. The new version of the file can be used in the same place as the old one with keeping the old version in the history of the software.

**Overview**: In Semantic MediaWiki, the admin can manage how different options in a page are shown, which people have access to the pages, which people can edit a page and etc. Creating pages,
adding information and editing is very user friendly in Semantic MediaWiki. But making tables, placing and sizing pictures and typing a text in different fonts and formats is not as easy as Microsoft Office. It might be better first the tables be created in excel file and then uploaded to the software.

**Authorization:** MediaWiki is community-driven, i.e. people who are registered in a wiki can create and modify documents on the wiki. There are some tools in Semantic MediaWiki that can control user rights, contributions and etc. In general, the wiki can be accessible for especial users, while registered users will have mostly access to all documents of the wiki and are able to make changes.
5 Findings and analysis

During setting up PFD has been tried to record design rationale and retrace it to design definition. Now the PFD contains both design definition and design rationale for the product Coromill490. The PFD is implemented in Semantic MediaWiki. On the other side at the company, the documentation for engineering knowledge is shared on an internal website.

In order to makes clear the pros and cons of the project results, the system functionality is discussed.

5.1 System Functionality

In the beginning of chapter 4 the requirements for a modelling system is mentioned. Based on these requirements the suggested system for the company can be using the concept of PFD, implementing in Semantic MediaWiki. This part of the project tries to evaluate the functionality of this system by considering the important parameters in knowledge modelling system:

**Capture:** Capturing information in PFD is based on design definition and design rationale. PFD contains every kind of information required for design engineering such as; text, picture, table and rule as well as the description and argumentation about them.

**Structure:** PFD is created by describing the main parts of the product in separate articles. Each article is linked to a page to record all the required information. The page can be linked to other pages or source of information. According to sub or super pages the information can be placed. Also documentation can be more classified by considering categories during recording the information. Categories are used as universal tags for articles, describing that the article belongs to a certain group of articles.

**Map:** Every page in Semantic MediaWiki contains a special part of PFD. The page can be linked to other pages or supported files. In this case it is easy to understand where the information is used.

**Store:** There is a need of a web server to store the information in Semantic MediaWiki.

**Search:** A good search provides a good accessibility to the knowledge. Searching for information in Semantic MediaWiki can be by typing a text and name of an article or using semantic search based on properties and query lines. The result of semantic search depends on the view of the person who has made the documentation. Considering search items during making documentation is necessary. Using properties might be confusing specially when different people are responsible for different parts of the document and they could give suggestions with personal focuses.
Findings and analysis

**Versioning:** Every page in Semantic MediaWiki can be edited. In the history of the page the previous changes of that page can be seen. Also a new version of a file can be uploaded and used instead of the old one by keeping the former file in the history of the software.

**Authorization:** In Semantic MediaWiki the registered users can modify the contents of a document. The admin can control the users and their contributions.

As it mentioned before creating document in Semantic MediaWiki is by typing text in the page or adding uploaded files. Typing text and designing tables in Semantic MediaWiki is not as easy as Microsoft Office. A large number of extensions and editors are available to be added to Semantic MediaWiki in order to improve the functionality of the system.

## 5.2 Evaluation the applicability and usefulness of PVM and PFD

Currently at Sandvik Coromant, the level of documentation can be affected according to the view of person who makes the documentation. Since there is a lack of a structured way for modelling documentation, product variant master (PVM) and product family description (PFD) are modelled at the company.

As it mentioned before, PVM shows a general view of the product according to the kinds and components of that product. This view helps the designer to have a better concept of the product structure. PVM tries to show the position of a part in the product according to the super or sub parts, which kinds of the part are available and what the related knowledge is such as; rules, tables, pictures and etc for that part. Since the reason of using PVM is to have an overall view, the detailed information of the product, the design rationale, might not be considered importantly. A way of showing PVM for a product family is product model manager (PMM) software. PMM by using the tree diagram could be a good way to represent this concept.

On the other side, the current documentation at the company contains just the design definition part of the engineering knowledge. In most cases people who were involved in a project should try to remember or discuss together about a rationale behind a rule or a feature. Since the design rationale is not recorded, reusing a previous solution in a new family is very difficult and time consuming. PFD solves this problem by documenting both design definition and design rationale. This documentation answers the question “why?” in addition of answering the question “what?” for a part of knowledge such as; text, rule, picture and etc.
5.3 Comparison of project findings and company documentation

The main part of this project is to set up PFD by using Semantic MediaWiki. PVM is introduced to the company to show the applicability of using it as general structure and if is possible, link it to PFD in Semantic MediaWiki. In Semantic MediaWiki using tree diagram is not common unlike PMM, although PFD can be created based on the main parts of PVM. In PFD each article explains a part class of the product. As it mentioned, PFD includes the argumentation and explanation regarding a rule, dimension, table, picture and etc (design rationale). In the current documentation at the company, design rationale is the lost part of knowledge. The thesis work has tried to show how capture design rationale and retrace it to design definition in order to create PFD.

Regarding the current documentation at Sandvik Coromant, recording the engineering knowledge is mostly in word or excel files. The files are sent to another department to share them on the company internal website. If the designer wants to make some changes he or she should edit the files and send them again to that department. This process is time consuming and human resource involving. In some cases the designer makes some changes without updating them on the website. By using Semantic MediaWiki, the knowledge is directly shared by just creating the pages and making documentation. Every page has a title and is linked to other pages or source of information. A page can be created, renamed, edited or deleted simply. The knowledge of a page would be edited and expanded by a group of authorized people who have access to the software. This people can argue about a case by making comments in the discussion part which is valid for every page. Semantic MediaWiki combines functions, editing and updating of the knowledge by editing a page and save it.

Semantic MediaWiki is free software and can be downloaded from its website, while using a web server should be considered by the company in order to use Semantic MediaWiki on the network.
6 Discussion and conclusion

Sandvik Coromant as a part of Sandvik Tooling is the world’s leading group producer of tools for turning, milling and drilling. For many years the company has worked with automation of their business processes and always been tried to improve it. There is continuously a need of new methods and tools to become more efficient in the product development process at the company. This project has tried to explore the concept of product family description at the company by working on a test product, Coromill490.

6.1 Discussion of method

First the project was started by a survey regarding general principles and applications. By having a look at chapter three, all the described methodologies and applications look interesting and they could be selected according to the objective of using the system. The important parameter for an application is to be user friendly and it becomes clear during implementation. The project has used two software, Product Model Manager and Semantic MediaWiki. Product Model Manager is free software for two months. Implementation in the software is very simple and there is no need for training. A manual description is available to help the user to know about the software. PMM is user friendly software and shows the concept of PVM in a good way. It would not be a good idea to expand the usage of software in other criteria instead of PVM, because of less flexibility. Since Sandvik Coromant needs one general application for storing a complete knowledge description of a product, PVM cannot be a good alternative as system modelling.

The main goal of the project is to set up PFD by using design definition and retracing design rationale. Since the available documentation for mostly products at the company contains just design definition part of the knowledge, design rationale was captured to set up PFD for Coromill490. The project of producing Coromill490 has taken place some years ago which could make it difficult capturing the design rationale after a while. The design rationale was recorded during meetings by the designer and discussing different issues related to a component or a part.

The PFD was created in Semantic MediaWiki. This step makes clear the applicability of both PFD and Semantic MediaWiki. Semantic MediaWiki is free software and can be downloaded from the website. The software was installed on the company’s computer. The manual help on the website completely explains the software and way of implementation. A new user needs some days to read the description in order to figure out how to run the software. Running the software for a new user is not difficult but there are many options, tools and information for learning, which could help the user to work in professional way. After making documentation as PFD, has been tried to make some
improvements in it. During improvement was tried to consider the designers and programmers view to expand the PFD.

6.2 Discussion of findings

PVM was the first product modelling of the project. PVM can be as an applicable concept at Sandvik Coromant by modelling the products based on parts and kinds of structure. The problem is that PMM doesn’t look like a desired application covers all required demands at the company. As an example, PMM doesn’t have a good search engine. On the other side, people at the company prefer to have one general system being used for all product families. Using two modelling for a product according to PVM and PFD would not be a good solution. However PFD can be created in an application based on part of structure as PVM.

During working on the project, has been tried to make the PFD according to design definition and design rationale for Coromill490. Also has been tried to link design rationale to design definition in the correct way. The PFD could be completed with more details and a better traceability if the description was created at the time of designing the product by the persons who were involved in designing. During improving the PFD has been tried to revise the description to make it simple and more qualified. The two examples show how to have a simple description, make information accessible, store the knowledge in the best place and make documentation in structured way. These activities are a part of the project objective which was met acceptable.

Semantic MediaWiki is introduced as an alternative application to implement PFD. The application should be user friendly and makes documentation accessible and provides reusing, expanding and editing the knowledge. In order to use Semantic MediaWiki, documentation should be done by considering categories and properties. In the project, the effort was to show some examples for these options but since just one product family was used for modelling, searching according to property or category might was not shown in the best way. By modelling more product families, more articles are created and the semantic search would be clearer.

6.3 Systematic evaluation

In order to gather the opinions of people regarding the results of the project at the company a systematic evaluation has been done with a number of questions discussed with four persons from different departments involved in the project. The persons are the design automation manager, the project leader for storing the knowledge in the company documentation, Coromill490 designer and a
Discussion and conclusion

system developer at Sandvik Coromant. The focus of the questions are on product variant master (PVM), product model manager (PMM) and the knowledge modelling system by creating product family description (PFD) in Semantic MediaWiki. The complete interview is attached to the report in appendix.

1. **What is your opinion regarding the concept of PVM?**
   
   *Answer:* The design automation manager believes that PVM is a good structure to model the product family. Also the product designer thinks PVM gives a good view of different parts and their relations together in a product family. But the project leader for documentation considers more the way of implementing PVM and connecting knowledge model documentation into design programming. In his opinion PVM cannot be integrated with the suggested modelling system.

2. **Can PVM be used at the company as a concept beside PFD?**
   
   *Answer:* It is an applicable structure for modelling product family which might be used at the company is the design automation manager’s answer to this question, while the project leader does not agree using PVM beside PFD. He prefers to have one modelling system using at the company. The product designer thinks PVM can be used at the company, although he prefers to first know the concept in more details.

3. **Can PMM be used at the company?**
   
   *Answer:* It could be used but it is not convenient using several applications at the company, the design automation manager says. The project leader is not interested in PMM software either. The product designer believes it could be a solution beside looking for more alternatives.

4. **What is your opinion about PFD?**
   
   *Answer:* People look satisfied regarding using PFD. Both the design automation manager and the project leader believe that PFD will be used at the company. The system developer thinks that PFD is essential for traceability, knowledge reuse and a streamlined parametric and rule-based process. The product designer says that PFD is more efficient than the current company documentation. Also he prefers to have an easy way to capture design rationale.

5. **How should design definition and design rationale be recorded?**
   
   *Answer:* The design automation manager says design definition and design rationale should be captured but according to the object it could be in different ways. It can be in the design modules with PMI or within the document (DMD or PFD). The project leader considers that people, especially the designers cannot start using a totally new system at the company. At this point the design information can be recorded in MS Word/Excel files with using templates. The product designer mentions time and believes design information should be captured during developing the product.
6. Can the designers according to their responsibilities create some parts of PFD?

*Answer:* The design automation manager answers yes and mentions that nowadays the designers are doing this job in the company documentation. Also the project leader and the product designer think that designers are the persons who should create PFD according to their responsibilities. The system developer believes PFD is the rule documentation artefact used by the design programmer in the development of the design program (PAD). Hence, the designer is solely responsible for creating and maintaining PFD.

7. Who should be responsible for managing the knowledge modelling system at the company?

*Answer:* Perhaps a group of people with good insight of development demands and sufficient knowledge about software and production is the answer of the product designer. The design automation manager, project leader and system developer have a common suggestion. It might be a department at the company which is involved in CAD, CAM and system supports.

8. What is your opinion regarding PFD as an input for design programming?

*Answer:* It is excellent. If it can be divided in Design Module Description, it’s easy to reuse and get the required knowledge for a family, the design automation manager says. Also the project leader agrees with the design automation manager that it is a good solution while the system developer thinks it should be mandatory to have a PFD as basis for the design programming. The product designer says PFD should be created during product development process with close collaboration with design automation.

9. What is your opinion about Semantic MediaWiki?

*Answer:* Easy to read, hard to write is a short but obvious answer of the design automation manager. The project leader looks interested in MediaWiki. But he is not sure how effective the semantic part will be when properties and categories are manually entered into each article. There is doubt about semantic part if it could be applicable in a big scale at the company. The system developer has a different opinion and he thinks there should be more work on Semantic MediaWiki to have a well-founded opinion about it. He thinks just modelling one product family with some examples is not enough to show the functionality of the software. Since the product designer does not have the experience of working by Semantic MediaWiki, he cannot judge the software but he is interested in search engine as well as editing the information and uploading quickly.

10. Do you prefer to use an application instead of using Microsoft Office for recording information?

*Answer:* The design automation manager says Microsoft Office is easy to work with, but also has some limitations. Sometimes using application is easier if it’s just needed to be filled in. Also he considers training for people regarding using a new application. Some designers prefer using Microsoft Office but of course the functionality of the application is an important parameter is the
Discussion and conclusion

project leader’s answer. The system developer agrees with the project leader that making a decision depends on the application. The product designer prefers using application with a good history to see all the previous changes within the documentation.

11. How do you think about making structured documentation based on categories?
   Answer: The system developer and the design automation manager have a common answer that using categories could be a way of structuring information. The product designer thinks structuring documentation based on categories is a bit difficult. The project believes structuring according to categories looks strange. It would be better and more applicable if properties and categories had some kind of automatic handling.

12. How do you think properties and categories can help people to have access to the stored knowledge?
   Answer: It is a way of searching and finding information is the design automation manager’s answer. The project leader says first documenting should be done in the right way to make accessible the information. The system developer answers the question in general and believes that a structured information model is essential to leverage reuse of knowledge. By having a structured information model it is possible to classify and search the information in a relevant way. The product designer mentions that using categories could be a way of accessing to information especially regarding reusing a solution in a new product family.

13. What is your opinion about making documentation in the pre-defined templates?
   Answer: The design automation manager and the project leader are those people who are very satisfied using templates but the project leader thinks that the way of using templates would depend on the system solution. A template makes it easy to find the desired information is the product designer’s answer. Also it could be a good help for new designers. The answer of the system developer to this question is that template is part of an information model. The template represents the class of the information and the document based on a template is an instance of that class. He sees template as an interface for people who use the information.

14. How can the results of the thesis work (PVM, PFD and Semantic MediaWiki) be used at the company?
   Answer: The design automation manager and the project leader unanimously say that PFD will be used at the company. They are not interested in using PVM. Semantic MediaWiki might be used completely or partly without semantic part. The system developer believes the important result of the work is to emphasize on the need to classify information. He does not judge Semantic MediaWiki but believes that application should support the creation and versioning of run able design rules and that is not possible with any known system today. But the technology is available for creating such a system which can be achieved by a long term goal. The product designer answers generally that the result of the project, partly or completely will be used during future works at the company.
15. What are the drawbacks of the project results?

*Answer:* The design automation manager would like to see more focus on Design Module Description as a part of the whole description parts. He thinks that people at the company can document Design Modules separately but by inheriting of PFD. Perhaps considering SharePoint beside the other applications during the survey is the project leader’s answer to this question. Also the product designer prefers to see the functionality of more applications and even modelling more than one product family.

16. What can be improved?

*Answer:* People more or less think that having a complete system solution can be the objective for improvement. The product designer prefers to evaluate the applicability of the system by modelling more than one product family.

17. Will the benefits outbalance the efforts and costs of resources required?

*Answer:* The project leader is not sure about the benefits but says that currently at the company there is a cost regarding finding information in unstructured and poorly documentation. He thinks if people make the documentation more efficient, there will be a saving of costs. By the way there is a need to visualise the current expenditures at the company. Regarding documentation, the design automation manager believes that since the philosophy of the company is to share and make accessible the knowledge for authorized people, documenting must be done in efficient way.

18. What is needed for further investigation?

*Answer:* The design automation manager wants to put more effort regarding the way of documenting and storing information in PFD while the project leader prefers to see how the designers can handle PFD in an efficient way. Again the product designer wants to know more software alternatives and select the most suitable one according to the company demands.

19. What would you like to see as next step?

*Answer:* Integrating the project results to the development processes at the company is the answer of the design automation manager. The project leader thinks that is better to focus on creating PFD, structuring documentation and etc. A general solution based on the results of the thesis work is the product designer consideration.

6.4 Conclusion and recommendations

The thesis work has been done in design programming department at Sandvik Coromant based on previous researches. The researches have tried to make documentation more efficient in design
automation. As it was shown in the company development processes, PFD is an input for design programming. Improving documentation in engineering design can effect and improve the results of design programming.

Since the company is in the beginning of using a knowledge modelling system, it would be difficult for people in different departments to use the whole results of this thesis work. The questionnaire evaluation shows that people are mostly satisfied to use PFD at the company. A good start of having a knowledge modelling system could be using PFD, describing the parts of the product family. Using templates to have the whole information for a desired component in PFD looks an interesting way to structure documentation. Creating PFD should take place during development process in order to make more efficient the documentation including all descriptions and argumentations regarding a section of information. For creating PFD design definition and design rationale should be captured. A recommendation could be making reports for design definition and design rational during developing the product.

When it comes to use an application, it is not fair to judge Semantic MediaWiki by modelling just one product family and showing two examples to evaluate the functionality of categories and properties. MediaWiki is a software with a wide range of usage. There would be efficient tools and extensions available to be added to Semantic MediaWiki for improving applicability of the software. SharePoint is another application used partly at Sandvik Coromant and the designers prefer to compare it to Semantic MediaWiki and then select the most applicable one. The best solution is to evaluate the advantages and disadvantages of SharePoint and Semantic MediaWiki in order to make a good decision.

In the end should be mentioned that there is a need to focus more in practical way during creating PFD in order to structure the description. Also integrating the project results to practical work for both engineering design and design programming would be another consideration which could be solved by further researches.
References

7 References

[3] Elgh, F., Sunnersjö, S., Cederfeldt, M., (2008), Strategies for adaptable design automation systems in the manufacturing industry, Project application, Jönköping University, School of Engineering
[4] Elgh, F., (2010), Design Automation, Jönköping University, School of engineering, Computer supported engineering design lecture, 10 May 2010
[18] Nilsson, M., Roadmap and design of YAPP CAD Turning tools, Sandvik Coromant, Sandviken, Sweden, 14 April 2010
8 Appendix

This appendix is about the interview with people from different departments at the company regarding the systematic evaluation.

Interview with the design automation manager at Sandvik Coromant:

1. What is your opinion regarding the concept of PVM?
   *Answer:* It is good to see the structure.

2. Can PVM be used at the company as a concept beside PFD?
   *Answer:* Well, the structure might be implemented, because it actually has been done.

3. Can PMM be used at the company?
   *Answer:* It could, but I rather think another system would be used. Team Centre is on its way, and I really think it could be implemented there, but I don’t know if it should. We do strive after as little extra software as possible.

4. What is your opinion about PFD?
   *Answer:* PFD is great. That’s a really good definition of our documentation at the company.

5. How should design definition and design rationale be recorded?
   *Answer:* It has to be recorded. I think it can be in different ways regarding what kind of “Why” it is. It can be in the design modules with PMI or in a document (DMD or PFD).

6. Can the designers according to their responsibilities create some parts of PFD?
   *Answer:* Yes! They should create documentation today. A PFD can be easier to create especially if it’s represented in a template (word, excel and etc.).

7. Who should be responsible for managing the knowledge modelling system at the company?
   *Answer:* CTS together with CTP. CTS responsible.

8. What is your opinion regarding PFD as an input for design programming?
   *Answer:* PFD is excellent. If it’s also divided in DMD, it’s easy to reuse and gets the needed knowledge for a family.

9. What is your opinion about Semantic MediaWiki?
   *Answer:* Easy to read. Hard to write.

10. Do you prefer to use an application instead of using Microsoft Office for recording information?
Answer: Office is easy to work with, but also has its limitations. Sometimes an application is easier if it just is needed to fill in things. On the other hand is something you have to use seldom and it can be hard to learn that application. I believe in templates, to make it easy.

11. How do you think about making structured documentation based on categories?
Answer: It can be a way. It’s like a template (for me).

12. How do you think properties and categories can help people to have access to the stored knowledge?
Answer: It makes the documentation searchable.

13. What is your opinion about making documentation in the pre-defined templates?
Answer: I like that.

14. How can the results of the thesis work (PVM, PFD and Semantic MediaWiki) be used at the company?
Answer: We will implement PFD and DMD. Probably not the other, at least not now. We have to see what our RBD project comes up with.

15. What are the drawbacks of the project results?
Answer: I don’t see any drawbacks. Although I would like to have more focus on DMD as a part of the whole Description parts. For me it feels that we can document Design Modules separately and by inheriting of various PFDs. That’s a way of reusing knowledge.

16. What can be improved?
Answer:

17. Will the benefits outbalance the efforts and costs of resources required?
Answer: Documentation should become better. We want company based knowledge to be able to spread that knowledge around to the employees.

18. What is needed for further investigation?
Answer: We will investigate a way (system?) to document and store documentation about PFD.

19. What would you like to see as next step?
Answer: Integrating the project results in our way of working.

Interview with the project leader of documentation at Sandvik Coromant:

1. What is your opinion regarding the concept of PVM?
Answer: I have trouble to see the real value with PVM with the system you have shown us. It means we have to create one “knowledge model” documentation that has no connection with the design program. In the end how do you connect PVM with the Design Programs? What will be the
advantages with PVM? I don’t see that. You also need to consider the competence of the designer that shall create and use the information.

2. Can PVM be used at the company as a concept beside PFD?
   *Answer:* I don’t think so, see question 1.

3. Can PMM be used at the company?
   *Answer:* No I don’t think so.

4. What is your opinion about PFD?
   *Answer:* Great! I like the concept and will work for a PFD solution at Coromant.

5. How should design definition and design rationale be recorded?
   *Answer:* At this point in MS Word/Excel files using templates. Again you have to consider where we are right now and if we implement a complete new system it will probably be a too large step for the designers (who will write the document).

6. Can the designers according to their responsibilities create some parts of PFD?
   *Answer:* Yes I think they should create the PFD documentation for their own product family.

7. Who should be responsible for managing the knowledge modelling system at the company?
   *Answer:* I know the answer but it is on-going work. It will be CTS but I cannot tell more right now.

8. What is your opinion regarding PFD as an input for design programming?
   *Answer:* I think it will be great.

9. What is your opinion about Semantic MediaWiki?
   *Answer:* MediaWiki is interesting. I’m not sure how effective the Semantic part will be when properties and categories are manually entered into each document. There are doubts about if it should work in a big scale.

10. Do you prefer to use an application instead of using Microsoft Office for recording information?
    *Answer:* It depends on the application. Some designers have said that they prefer Office products.

11. How do you think about making structured documentation based on categories?
    *Answer:* Strange. If it would be possible to have it more in forms then it would be better. I mean that the properties and categories should have some kind of automatic handling.

12. How do you think properties and categories can help people to have access to the stored knowledge?
    *Answer:* Implementing in the right way would be helpful for information finding. But see question 11.

13. What is your opinion about making documentation in the pre-defined templates?
Great, I like it a lot and we will use templates in the future. In which way depends on the system solution.

14. How can the results of the thesis work (PVM, PFD and Semantic MediaWiki) be used at the company?
Answer: PFD will definitely be used. PVM will not be used. Semantic MediaWiki is under consideration but mostly without the semantic part.

15. What are the drawbacks of the project results?
Answer: Maybe should have been considered having look at SharePoint.

16. What can be improved?
Answer: A complete system solution.

17. Will the benefits outbalance the efforts and costs of resources required?
Answer: I’m not sure what the benefits are. But if we look at PFD we have a cost today regarding finding information in unstructured and poorly documentation. From my point of view, we must make the documentation process more efficient and therefore there will be a saving and no costs. We just have to visualise the costs we have today.

18. What is needed for further investigation?
Answer: How can the designer handle the PFD in the most efficient way?

19. What would you like to see as next step?
Answer: Which is the best way to create PFD? How to structure documents and so on.

Interview with the system developer at Sandvik Coromant:

1. What is your opinion regarding the concept of PVM?
Answer:

2. Can PVM be used at the company as a concept beside PFD?
Answer:

3. Can PMM be used at the company?
Answer:

4. What is your opinion about PFD?
Answer: See answer for question 6. The PFD is essential for traceability, knowledge reuse and a streamlined parametric and rule-based process.

5. How should design definition and design rationale be recorded?
Answer: See answer for question 14.
6. Can the designers according to their responsibilities create some parts of PFD?
   **Answer:** As far as I understood regarding the concepts of PID, PFD and PAD, PFD is the rule documentation artefact used by the design programmer in the development of the design program (or PAD). Hence, the designer is solely responsible for creating and maintaining the PFD.

7. Who should be responsible for managing the knowledge modelling system at the company?
   **Answer:** This is the responsibility of CTS and its sub-departments.

8. What is your opinion regarding PFD as an input for design programming?
   **Answer:** It should be mandatory to have a PFD as basis for the design program.

9. What is your opinion about Semantic MediaWiki?
   **Answer:** See answer for question 14.

10. Do you prefer to use an application instead of using Microsoft Office for recording information?
    **Answer:** See answer for question 14.

11. How do you think about making structured documentation based on categories?
    **Answer:** Properties is one way of making the information structured. A structured information model is essential to leverage reuse of knowledge. By having a structured information model it is possible to classify and search the information in a relevant way.

12. How do you think properties and categories can help people to have access to the stored knowledge?
    **Answer:** See answer for question 11.

13. What is your opinion about making documentation in the pre-defined templates?
    **Answer:** This is part of an information model. The template represents the class of the information and the document based on a template is an instance of that class. I see template as a contract or interface for consumers of the information (and by consumers I mean humans or other systems).

14. How can the results of the thesis work (PVM, PFD and Semantic MediaWiki) be used at the company?
    **Answer:** The important result of the work is to emphasize on the need to classify information. I haven’t seen enough of Semantic MediaWiki to have a well-founded opinion about it, but as I see the future of PFD’s and design rule documentation I would say that Semantic MediaWiki does not suffice. In the long term the PFD recording application should support the creation and versioning of runnable design rules and that is not possible with any known system today. But the technology is available for creating such a system and I think that should be the long term goal.

15. What are the drawbacks of the project results?
    **Answer:**
16. What can be improved?
   Answer:

17. Will the benefits outbalance the efforts and costs of resources required?
   Answer:

18. What is needed for further investigation?
   Answer: See answer for question 14.

19. What would you like to see as next step?
   Answer: See answer for question 14.

Interview with Coromill490 designer at Sandvik Coromant:

1. What is your opinion regarding the concept of PVM?
   Answer: What I have seen PVM is a good way of visualizing how different parts of a product family are linked together and what is included.

2. Can PVM be used at the company as a concept beside PFD?
   Answer: My knowledge about PVM is not enough but I can’t see why not.

3. Can PMM be used at the company?
   Answer: It could be a solution, but at this stage we look in to several possibilities.

4. What is your opinion about PFD?
   Answer: The current documentation at the company describes our product families. PFD is perhaps a better description for that in the future. Also if we can implement the rationale part of this in an easy way it could be great.

5. How should design definition and design rationale be recorded?
   Answer: The best way is during development if it is connected to the function of the concept. If it is the rationale for a specific part of the design or sub part, it should be recorded in the documentation for that at least when the design is frozen.

6. Can the designers according to their responsibilities create some parts of PFD?
   Answer: I can’t see anyone else doing it, but of course in close cooperation with design automation.

7. Who should be responsible for managing the knowledge modelling system at the company?
   Answer: Perhaps a department with good insight in the needs from the development and other departments related to product preparation. Also with good knowledge about software and our way of working.
8. **What is your opinion regarding PFD as an input for design programming?**  
*Answer:* If the PFD is created during PDP in close cooperation with design automation it generates a good input for the design programming.

9. **What is your opinion about Semantic MediaWiki?**  
*Answer:* Since my knowledge about this software is very basic, the strengths towards other applications are unknown to me. But the little I have seen shows that there are possibilities regarding how a user can set up different search criterion. I like the idea that you can edit and upload quickly.

10. **Do you prefer to use an application instead of using Microsoft Office for recording information?**  
*Answer:* If there are good templates and stored in a way that history of changes can be seen, than the Microsoft Office is ok for now. As a company I think we should strive to minimize the number of software.

11. **How do you think about making structured documentation based on categories?**  
*Answer:* You should always strive to make the documentation in a structured way. It could be difficult to strictly document based on properties and categories. During documenting a product family I think you have to make sure that there is a good overview what is included in that particular family.

12. **How do you think properties and categories can help people to have access to the stored knowledge?**  
*Answer:* When designing a new product you can look for what is already made in a certain area. Can you reuse it?

13. **What is your opinion about making documentation in the pre-defined templates?**  
*Answer:* With pre-defined templates you are helped during making documentation, easy to see what is expected from you. For new designers it should be great, and also the format for the documentation will be more streamlined.

14. **How can the results of the thesis work (PVM, PFD and Semantic MediaWiki) be used at the company?**  
*Answer:* It can be a source of information and insight during the future work within rule based design at Sandvik Coromant.

15. **What are the drawbacks of the project results?**  
*Answer:* If there had been more time perhaps looked into other software and evaluated those as comparison with the ones chosen in the thesis. Also look into more product families.

16. **What can be improved?**  
*Answer:*

17. **Will the benefits outbalance the efforts and costs of resources required?**  
*Answer:* I don’t know.
18. What is needed for further investigation?
*Answer:* To look into more software alternatives and decide most suitable for our requirements.

19. What would you like to see as next step?
*Answer:* To use the outcome of this thesis as I mentioned in question 14, the best possible way.