INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED11
15 - 18 AUGUST 2011, TECHNICAL UNIVERSITY OF DENMARK

MODELING AND MANAGEMENT OF PRODUCT KNOWLEDGE IN AN ENGINEER-TO-ORDER BUSINESS MODEL

Fredrik Elgh
Jönköping University

ABSTRACT

To adopt an engineer-to-order (ETO) business model when competing on a market where competitors' products are mass-produced is a challenge. However, a competitive edge can be gained if the principles of ETO and mass production successfully can be combined. High level of customer adaptation requires systems for efficient generation of product variants with associated specifications for automated manufacturing. To maintain these systems's usefulness over time, frequent updating will normally become a necessity. Of equal importance, is the reuse of the system encapsulated generic product family objects when developing a new product family. In this paper a case study is presented with the main objective to provide a system foundation for modeling and management of product knowledge supporting reuse, expansion and maintenance of system embedded objects. One of the central parts of the framework is the Meta-Knowledge Containers, labeled Descriptions for the case company. A Description contains both a definition of system embedded objects as well as the rationale behind their design. Traceability is gained by linking of Descriptions, individually and to documents, models and items.

Keywords: Engineer-to-order, knowledge management, knowledge modeling, design automation

1 INTRODUCTION

Many companies base their business strategy on customized products. To enable such a strategy they make use of advanced application systems for automating the work of generating product variants based on different customer specification, i.e. they develop and use design automation systems. This technology then becomes not only a means of improved efficiency but also a method for drastically reduced lead times, improved offer precision, quality assurance, performance and a higher degree of customer adaptation. The establishment of a design automation system is a significant investment in time and money which is expected to give revenues over many years. For a design automation system to maintain its usefulness over time, frequent updating of design rules and execution control will normally become a necessity. Experience indicates that 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. Reuse of the system encapsulated generic product family descriptions, for example design rules, when developing a new product family is perceived to significantly increase the efficiency in system development and is a means to reduce the market introduction lead time. The scope and the purpose of this research originate from industrial problems and needs which have been identified within research projects carried out in near collaboration with industrial partners. The starting-point in industrial problems is in accordance with problem-based research as described by Jørgensen [1] and Blessing's research methodology for the development of design support [2]. The introduction, evaluation, and refinement of new concepts that are perceived as prescriptive models are in accordance with the design modeling approach [3]. The focus of this paper is a case study carried out at a company with long experience of systems for automated variant design. The main objective is to provide a system foundation for modeling and management of product knowledge supporting reuse, expansion and maintenance of system embedded knowledge.

2 RELATED WORK AND STATE OF THE ART

For this work the contributions in the domains of product configuration, design automation and knowledge based engineering (KBE) are most relevant as they are focused on specific applications. Hvam et al [4] describes a complete and detailed methodology for constructing configuration systems in industrial and service companies. They suggest an iterative process including the activities: analysis of product portfolio, object-oriented modeling, object-oriented design and programming, among others. Every activity results in a description of the problem domain with different levels of abstraction and formalization. Two strategies are proposed for system documentation, either by using a product variant master and associated CRC (Class Relationship Collaboration) cards or by using the class diagram of a formal model and associated CRC-cards. The original content and structure of the CRC-cards have been further developed by Haug and Hvam [5] and Haug et al [6] presents a prototype system for the documentation of the CRC-cards, the product variant master and the class diagrams. Regarding design automation systems, a procedure for development of has been outlined by Rask [7] where issues about documentation and maintenance are addressed by emphasizing the need and importance of routines regarding versioning, verification and traceability. A possible means to support the updating of the knowledge-base proposed is to strive for a design automation system implementation that allows the revision and the documentation to be executed at system runtime [8]. The development of knowledge based engineering applications is supported by MOKA (Methodology and software tools Oriented to Knowledge Based Engineering Applications) [9]. Two central parts of the methodology are the Informal and Formal models. The Informal model is used to document and structure knowledge elicited from experts, handbooks, protocols, literature etc. The Formal model is derived from the Informal model with the purpose to model and structure the knowledge in a fashion suitable for system specification and programming. In the area of platform-based product development the concept of configurable components introduced [10]. One element of the configurable components is a function-means model set including functional requirements, constraints, and design solution. The purpose with the function-means model is to provide design rational for the encapsulated design solutions. This could support the understanding of the system and thereby enhance system adaptation and maintenance. The issue with providing traceability and design history documentation for products generated by the use of executable design algorithms is discussed in Sunnersjö et al. [11]. A system is proposed to be based on files incorporating design knowledge and executable statements. These files are managed by a PDM system and they are manually executed for the creation of design variants based on different customer specifications. Principles for an automated execution using a predefined workflow based on the Dependency Structure Matrix (DSM) method is presented by Sunnersjö et al. [12] and a model for management of manufacturing requirements is presented by Elgh and Sunnersjö, [13]. The subject was further explored and Elgh [14] introduce principles for the modeling and management of manufacturing knowledge in design automation systems with an associated information model. The information model incorporates default links and runtime created links between manufacturing requirements, manufacturing resources, knowledge objects [15] and product items. The knowledge objects include pointers to the implementation of the knowledge (e.g. a spreadsheet file or a parametric CAD file). The principles were applied when developing a prototype system for automated variant design.

As can be concluded from the above, some research has been conducted in the area of system development in general and on the modelling of product related knowledge and information in specific. In this work, the starting point has been the state of practice and the identification of problems and needs related to modelling and management of product related knowledge at a company. A conceptual framework and a principle solution have been developed and in a later stage applicable theories and models provided above will be considered for system realisation.

3 INITIAL CASE STUDY

Information about the case company was gathered by meetings, demonstrations of applications, reviews of documents and in-depth interviews. Eight respondents with different company positions were interviewed using a standardized questionnaire with open-ended questions. The result from the case study includes a description of the company’s means of providing special products at the same cost as for standard products. Further, the documentation and the management of product related knowledge at the company are revealed followed by an analysis and a problem discussion.

3.1 Business model and means for custom engineered products

The company develops and manufactures products for the mechanical industry. It is a global company represented in many countries worldwide. Manufacturing is located at several production units and for customer support there are a number of productivity centers. The product catalogues with standard

products contains ten thousands of articles. Each individual product structure is not complex but a large number of variants exist and the catalogues contain only the most frequent variants. It is of vital importance for the company to, beside the standard products, provide special products based on different customer demands. These custom engineered products represent an essential part of the delivered products. A request for quotation of a custom engineered product is guaranteed to be replied within 24 hours, including design drawings and a final price. All the necessary documents and manufacturing programs are automatically generated when the bid is accepted. The product space also includes products that are supported by manufacturing and special engineered products.
Automation of different activities started at 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 paths to CNC machines), steer information to production cells, and measuring preparation (creation of programs to CMM machines). The automation of these different activities has resulted in a stream-lined process for quotation and order preparation, Figure 1.

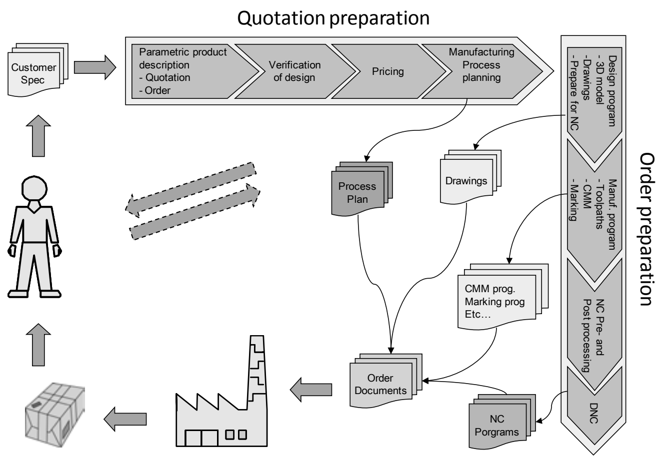

Figure 1. Automated process for quotation and order preparation

By means of in-house developed applications, the process is executed automatically requiring no manual interaction. This enables the company to provide custom designed tools rapidly and efficiently in the same way as standard tools. The quotation and order process includes automatic:

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

3.2 Company development process

The principle development process deployed at the company is depicted in Figure 2.

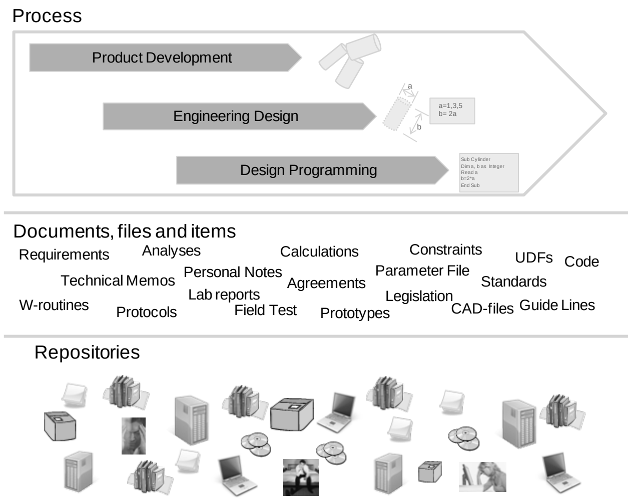

Figure 2. Company development process, information and repositories

The process differs from a traditional product development process as it is aimed at describing a product space by rules and digital models, starting with marketing and ending with application programs. In the development task, individual instances of a planned product family is developed and verified, including, by example, structural analysis, functional tests, CAD modeling and building prototypes. Based on these instances, a design space of the product family is defined and described by rules and associated 3D solid models. The rules are documented and structured as expressions, tables and figures. The rules are required input for the design programmers who prepares the 3D models with information (e.g. geometries, datum features and named surfaces) to be used when creating programs for product design as well as programs for CAM and CMM. Design programming also includes the adaptation of the product family description for the system used for application development, actual coding in that system and verification of application. The output is what can be called a Design Automation Model (Figure 3) incorporating two sub-models, a Rule-model and a CAD-model, defined by a number of different model elements (building-blocks).

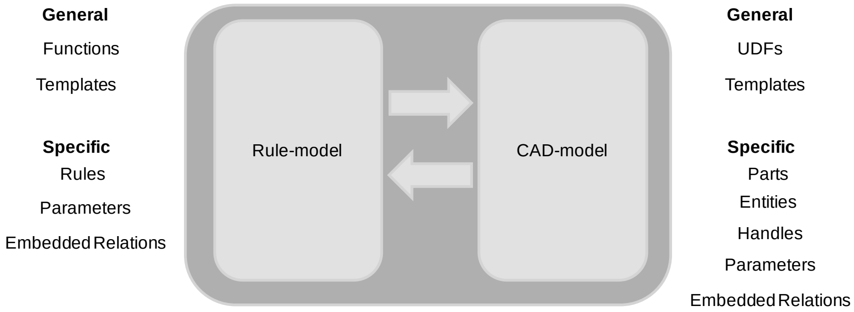

Figure 3. Building-blocks of the Design Automation Model.

The principles are the same for CAM and CMM programming. CAM and CMM programming are preceded by CAM and CMM preparation. The three applications of executable product and manufacturing descriptions, describing a solution space – not just single instances – are loaded into a main system that manages the execution process of individual quotations and orders.

One essential means used in the quotation and order preparation is the design programs for the different product families. The outputs from the design programs includes variant specific: 3D models configured for CAM preparation and CMM preparation; quotation drawings, assembly and manufacturing drawings; customer data (e.g. drawings and 3D model).

A design program consists of a number of rules that are executed in a sequence resolved by an inference engine. The design programmers have during the years used different systems for rule specification. The current environment is developed at the company and includes a rule engine. Currently, the company is working on the next generation of programming environment which will be base on an object oriented approach.

3.3 Documentation and management of product related knowledge

In this study the identified documentation was classified as either product related or process related. Product related documentation includes: object documentation describing the result of an activity (e.g. a description of a parametric CAD model), object process documentation which describes the work related to the object (e.g. considerations, tests, analysis, decisions, assumptions etc.), and guide lines regarding the product design considering specific aspects (e.g. manufacturing and environment). Process documentation, on the other hand, is related to a specific product development project including documents for project management, meeting protocols and other documents used for sharing information between project members. Process documentation is stored and managed using a project database. Object documentation and guidelines are stored, managed and published on a company internal portal. The design engineers provide the material regarding the product design that is to be published on this portal. No central system for storing and managing object process document exists to date, however, some individuals make notes in documents or in programming code for personal use or to be used by other group members. An overall summary of the in-depth interviews gives that:

- The purpose of documentation in general and project documentation in specific is not seen by all company employees.

- The quality of the documentation is very varying.
- The corporate project database is used for finding work prerequisites and to learn from earlier projects.
- It is perceived, by the respondents, hard to find project documents for non project members.
- The information in the corporate project database is coarse and not easily accessible for non project members, especially when the project has been closed. The system is mainly used to find specific individuals for consultation regarding, by example, reuse of product descriptions.
- The documents are weakly connected to the different product families.
- Specific geometries, CAD models, are reused to some extent but design rules and principles are seldom reused.
- It is difficult for individuals who has developed good solutions to share these solutions. The reason given for this is that there is no present system for such documentation.
- The access to information is seen as most difficult by design engineers and design programmers.
- Actual testing is used as method for checking validity and quality when reusing different types of product descriptions in an application.
- A general view is that reuse could be augmented at the company and improved documentation could support this. However, it is important that documentation can be easily done.

Current documentation at the company is mainly focused on describing the final results of different activities answering “What?” questions. To reuse a rule in a new context (another product family) requires more information, for example scope, range, simplifications and underlying assumptions. Such information might be enough if the rule is to be used as it is, but if the rule has to be modified and adapted to specific circumstances even more information is required to support the adaptation while ensuring the validity of the rule. Documentation is perceived as important but commonly viewed as a non-value adding activity in the development projects due to:

- The purpose, objectives and users of specific documentation is not defined and communicated.
- The long-term corporate value can be difficult for individuals to grasp
- Lack of time
- Lack of support

4 CONCEPTUAL FRAMEWORK AND PRINCIPLE SOLUTION

It can be concluded from the result of the initial case study that there is a general need of principles and methods to support capturing of knowledge and knowledge traceability at the company. In this chapter, a conceptual framework for modeling and management of product knowledge in an engineer-to-order business model will be presented together with the principle solution for the case company.

4.1 Tasks, domains and general enablers

Tasks to be supported can be related to two domains concerning either the family specific building blocks, i.e. rules and CAD-models, or the general building blocks, i.e. functions, UDFs and templates. Three different tasks have been identified as essential to support and these are reuse, expansion and maintenance. Reuse is the use of existing building-blocks in a new context (e.g. a new product family or system foundation). Currently two different strategies can be identified at the company depending on the domain of application: product family, use a previous solution or create general solutions (e.g. functions, UDFs); system foundation, system independent platform with system interfaces. Expansion implies increasing the design space or functionality (e.g. scale the parameters’ ranges or extend the topology). The strategy adopted at the company is to manually investigate if necessary changes can be implemented without violating any other conditions. Maintenance concerns modifying existing family specific building blocks or general building blocks according to new circumstances (e.g. changes in manufacturing constraints, material properties, manufacturing processes, legislations, standards etc.). The strategy adopted at the company is to manually investigate if necessary changes can be implemented without violating any other conditions. In addition to the domains and the tasks identified above, three general enablers for successful task execution are the structuring, the validation and the adaptation of model elements. Structuring is required for the purpose of enabling searching in pursuit of finding candidate building blocks for reuse, expansion or maintenance. Validation is required to ensure the applicability of candidate building blocks. Adaptation is necessary when changes are required to make the selected building blocks applicable in a new context.

4.2 Model elements and knowledge

Initially, the focus will be on the domain of family specific building blocks. Looking at these objects and especially how rules relate to the concept of knowledge, where knowledge is seen as an intentionally defined element that systematically transforms input to output, leads to the conclusion that rules are a kind of knowledge. Commonly, rules implements computations, actions, consequences and relations but they do not encapsulate the argumentation for their existence or the reason behind their design. The definitions of rules are based upon insights, decisions or facts derived from prerequisites, trial and error, experience, calculations, simulations, experiments, filed tests, literature etc. which constitutes another kind of knowledge that can provide a deeper understanding of the rules. A deeper understanding of rule can be supported by the answers to questions such as Why, When, Scope, Valid ranges of input/output, Origin, Supporting theories, Simplifications, Assumptions etc. The answers to these questions constitute knowledge about knowledge i.e. Meta-Knowledge (Figure 4) and the rules, the UDFs and the parametric CAD-models are different types of Knowledge Objects.

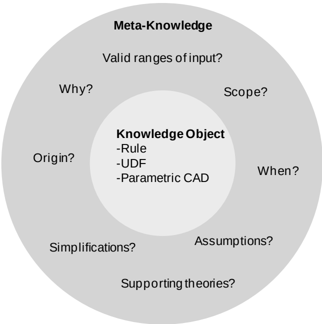

Figure 4. Knowledge Object and Meta-Knowledge

In the product development process knowledge are processed in different steps and appears in different states. At the company, six knowledge states can be identified and these are: unprocessed, selected, structured, adapted, utilized and documented (Figure 5). To support reuse, expansion and maintenance of different Knowledge Objects (building blocks) it is required that the focus in the product development process is not limited to the definition of Knowledge Objects exclusively, but also include the definition and collection of associated Meta-Knowledge.

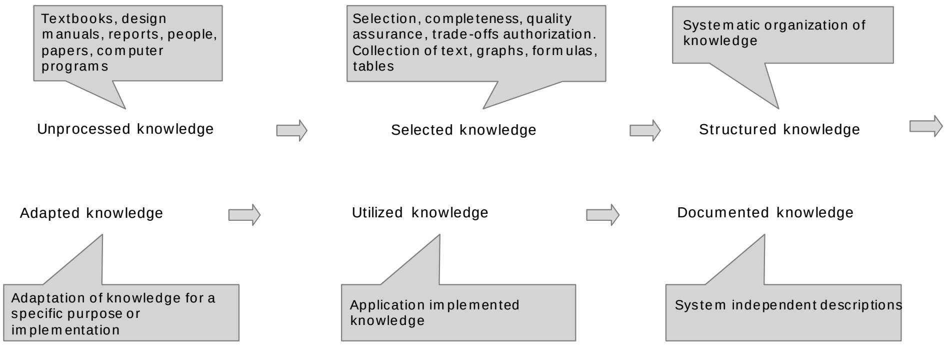

Figure 5. Knowledge states

When analyzing the company development process (Section 3.2 *Company development process*) potential and relevant Meta-Knowledge can be identified as appearing in different objects (e.g. documents, models and items) stored in different locations (repositories). These objects, labeled Meta-Knowledge Carriers, are generated throughout the development process to support the definition of Knowledge Objects. However, there is no mapping between the output from a sub-process and supporting documents, files and items (Meta-Knowledge Carriers) and no description that provides a selection, a context and a meaning to the content of the Meta-Knowledge Carriers in respect to the output from the sub-process. Further, there is no traceability between the processes: Product Development, Engineering Design and Design Programming. The objective is to tackle these deficiencies by the introduction of what could be generally called Meta-Knowledge Containers. Meta-Knowledge Containers provides mapping of, and meaning to, individual Meta-Knowledge Carriers as depicted in Figure 6.

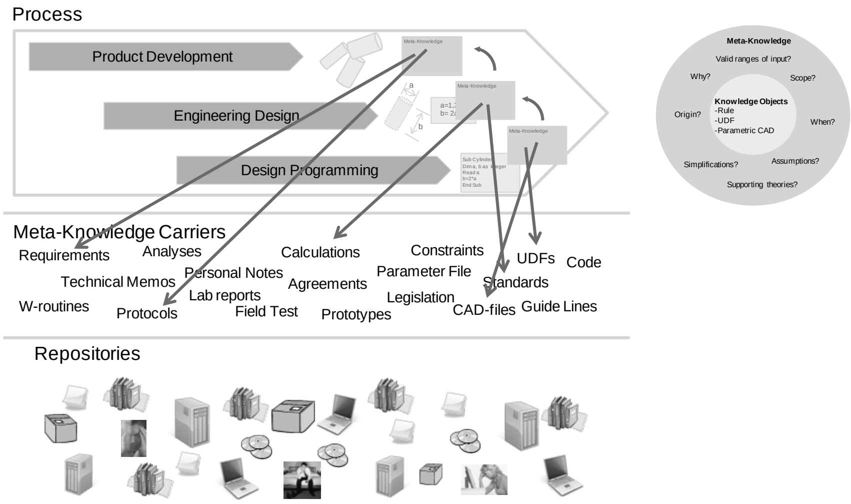

Figure 6. Principle solution with Meta-Knowledge Containers.

4.3 Descriptions

In this specific case, the concept of Meta-Knowledge Container was labeled Description as it suited the existing terminology at the company. The concepts of Design Definition and Design Rational were also introduced, where Design Definition is the Object documentation and the Design Rational is the Object process documentation (see Section 3.3 *Documentation and management of product related knowledge*). The main focus of the Design Definition is the construction and the function of a process output object whereas the main focus of the Design Rationale is the argumentation and supporting descriptions unfolding and justifying the object design. Both the Design Definition and the Design Rationale provides essential meta-knowledge about the process output object and together they constitute the foundation for the Design Description. For the case company, three different Design Descriptions related to the three main processes of the development process were introduced to enable documentation and traceability (Figure 7):

- Product Development, Product Instances Description (PID)
- Engineering Design, Product Family Description (PFD)
- Design Programming, Product Automation Description (PAD)

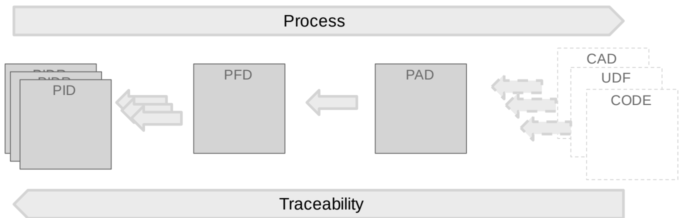

Figure 7. Company introduced Descriptions.

In general, the three Descriptions serve as collectors. Main function and properties are:

- Support capturing of Design Definition and Design Rationale
- Links to supporting documents, models and items
- Links to preceding Descriptions
- Written for a clearly defined purpose and potential users
- Based upon templates with predefined headings, keywords and fields
- Simple and visual
- Continuously updated
- Versioning control
- Authorization functions
- Has an owner

The intention with the templates is to facilitate the work of documenting and support high quality documentation. The content of a Description include, by example, an explanation of the overall product and its building blocks at different levels (e.g. product, assemblies, parts, features and geometrical entities), relations between building blocks (e.g. functional structure and assembly sequence), parameters (input, internal and output), and rules describing the design space. This will constitute the Design Definition of a Description. By adding information and links concerning aspects such as calculations, analyses, field test, underlying principles for design, assumptions, constraints, context, valid ranges of parameters and aspects for validity of rules, together with statements regarding what to consider when changing, ideas not yet implemented and workarounds, the Design Rational of a Description is completed. Means for information representation include tree models, text, illustrations, pictures, tables, formulas, links and meta-data. The scope and content of the three Descriptions introduced at the case company is illustrated in Figure 8.

- PID
  - Implementation independent
  - What, why, …?
  - Product instances architecture
  - Assemblies
    - Order
    - Placement
    - …
  - Parts
    - Features
    - Materials
    - Dimensions
    - Tolerances
    - Surface roughness
    - Properties
    - …
  - Pointing at supporting documents and items
  - Purpose
    - Support reuse, expansion and maintenance
  - Users
    - Product development
    - Engineering Design

- PFD
  - Implementation independent
  - What, why, …?
  - Product family architecture
  - Configurations
  - Assemblies
    - Order
    - Placement
    - …
  - Parts
    - Features
    - Materials
    - Dimensions
    - Tolerances
    - Surface roughness
    - Properties
    - …
  - Rules
    - Expressions
    - Tables
    - …
  - Elements
    - UDF's
    - CAD-models and drawings
    - CMM and CAM entities
    - …
  - Pointing at supporting documents, items and statements in PID
  - Purpose
    - Support reuse, expansion and maintenance
  - Users
    - Engineering Design
    - Design Programming

- PAD
  - Adapted for implementation
  - What, why, …?
  - Product family architecture
  - Configurations
  - Assemblies
    - Order
    - Placement
    - …
  - Parts
    - Features
    - Materials
    - Dimensions
    - Tolerances
    - Surface roughness
    - Properties
    - …
  - Rules
    - Expressions
    - Tables
    - …
  - Elements
    - UDF's
    - CAD-models and drawings
    - CMM and CAM entities
    - …
  - Pointing at supporting documents, items and statements in PFD
  - Purpose
    - Support reuse, expansion and maintenance
  - Users
    - Design Programming

Figure 8. Scope and content of PID, PFD and PAD

4.4 Overall solution and information model

An outline of the overall solution aimed at supporting reuse, expansion and maintenance is depicted in Figure 9. Process output objects (e.g. Knowledge Objects), PID, PFD, PAD, Knowledge Carriers (e.g. project documents, models and items), meta-data and links are stored in a database managed by a Database Management System. System functionality includes means for capture, structure, map, store, retrieve, search, visualize, versioning and authorization.

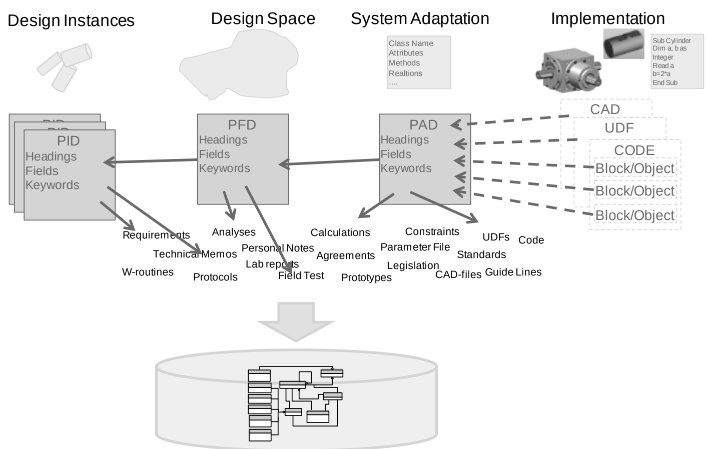

Figure 9. Overall solution

Of essential importance for system functionality is the information model (Figure 10). As an example the traceability of a code statement is illustrated. The code is to be divide into collections of statements or objects that is linked to a PAD statement. The PAD statement is liked to a PFD statement as well as to applicable Meta-Knowledge Containers. The idea is to work with complete Descriptions to enable completeness, overview and context to support understanding of individual statements but also provide accurate meta-knowledge with high granularity by the subdivision into statement that can be linked. Reviewing the need, concepts and functions of a system implementation ends in a solution that combines the principles of product variant master with the functions of a PDM-system. To enforce the importance of documentation and traceability and to ensure that completeness and validity of descriptions are maintained will require the introduction of a new role – a Descriptions Manager. The Descriptions Manager will act as an integrating function across sub-processes and departments responsible for all descriptions during the whole product family life-cycle.

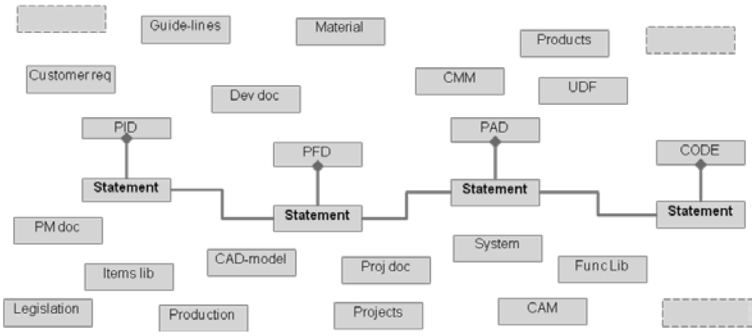

Figure 10. General information model

4.4 Company response

The conceptual framework and principle solution have evolved as part of a research project conducted in near collaboration with the company. The presented result has been discussed with six company representatives representing design programming and system development in a two-day long meeting. Aspects discussed included overall impression, missing parts, out of scope, concepts and terminology, system functionality, merging of Descriptions, merging of roles, Descriptions process, Description Manager and critical parts for function. The company response was positive. The concept of Descriptions was perceived as meeting the needs of capturing, structuring and linking knowledge. The importance that documentation must be done continuously was expressed. Guidelines for documentation during product development projects are requested and it should preferably be possible to document in the systems you are working in. The idea supports a cross product family focus beyond the actual project which is even further enforced by the Descriptions Manager. Implementation in existing PDM-system was not perceived as a possible solution at the company. However, as responded by a system developer “Everything is possible to implement as we are developing our own systems”. To remember though, is that a critical part for successful implementation is a strong support and understanding among all involved parties in the organization.

4 CONCLUSION

The main objective of this work was to develop a system foundation for modeling and management of product knowledge supporting reuse, expansion and maintenance of design automation system embedded generic product family objects. In this paper, a conceptual framework and a principle solution have been outlined and discussed. One of the central parts of the framework is the Meta-Knowledge Containers, labeled Descriptions for the case company. Descriptions are to be created for identified sub-process and delivered together with the main output. A Description should contain both a definition of the output as well as the rationale behind its design. Traceability is supported by linking statements between descriptions and to Meta-Knowledge Carriers. Initial discussions and evaluations at the company confirm the applicability and usefulness of the proposed approach. Of central importance is that the benefits gained outweigh the effort required. Unfortunately, this can be hard to prove by investment calculations for this kind of system. To fully validate the presented explorative work and its feasibility entail future studies and improvements that requires the development of a pilot system. This will be subject for future work.

ACKNOWLEDGEMENTS
This work was conducted within a Swedish Knowledge Foundation granted project and financial support is gratefully acknowledged. The author would also like to express his gratitude to the company for providing information and knowledge as well as for collaboration and helpful discussions.

REFERENCES
[1] Jørgensen K. Paradigms for research work. *Lecture notes*, 1992 (Institute for Production, Ålborgs University Center, Ålborg).
[2] Blessing L. *A Process-based approach to computer-supported engineering design, Ph.D. thesis*, 1994 (University of Twente, Twente).
[3] Duffy A. and Andreasen M.M. Enhancing the evolution of design science. In *Int Conference on Engineering Design*, ICED'95, Vol. 1, Praha, August 1995, pp. 29-35 (Heurista, Zürich).
[4] Hvam L., Mortensen N.H. and Riis J. *Product customization*, 2008 (Springer Verlag, Berlin).
[5] Haug A. and Hvam L. CRC-cards for the development and maintenance of product configuration systems. In *Joint Conference IMCM06 and PETO06*, Hamburg, June 2006, pp. 369-381 (GITO-Verlag, Berlin).
[6] Haug A., Degn A., Poulsen B., and Hvam L. Creating a documentation system to support the development and maintenance of product configuration systems In *WSEAS Int Conference on Computer Engineering and Applications*, Gold Coast, January 2007, pp.122-131.
[7] Rask I. *Rule-based product development - Report 1*, 1998 (Industrial Research and Development Corporation, Mölndal)
[8] Rask I., Sunnersjö S. and Amen. R. *Knowledge based it-systems for product realization*, 2000 (Industrial research and development corporation, Mölndal).
[9] Stokes M. *Managing engineering knowledge – MOKA*, 2001 (Prof Eng Publications ltd, London).
[10] Claesson, A. *A configurable component framework supporting platform-based product development, Ph.D. thesis*, 2006 (Chalmers University of Technology, Gothenburg).
[11] Sunnersjö S., Rask I., and Amen R. Requirement-driven design processes with integrated knowledge structures. In Proceedings of DETC/CIE, Chicago, September 2003, pp 439-449 (American Society of Mechanical Engineering, New York).
[12] Sunnersjö S., Cederfeldt M., Elgh F. and Rask I. A transparent design system for iterative product development, *Journal of Computing and Information Science in Engineering*, 2006, 6(3), pp. 300-307.
[13] Elgh F, Sunnersjö S. An ontology approach to collaborative engineering for producibility, *Int Journal of e-Collaboration*, 2007, 3(4), pp. 21-45.
[14] Elgh F. Supporting modelling and management of manufacturing knowledge in design automation systems, *Advanced Engineering Informatics*, 2008, 22(4), pp. 445-456.
[15] Elgh F. and Cederfeldt M. Concurrent cost estimation as a tool for enhanced producibility – system development and applicability for producibility studies, *Journal of Production Economics*, 2007, 109(1-2), pp. 12-26.

Contact: Fredrik Elgh
Jönköping University
Department of Mechanical Engineering
Gjuterigatan 5
551 11, Jönköping
Sweden
Tel: Int +46 36 101672
Fax: Int +46 36 125331
Email: fredrik.elgh@jth.hj.se

Fredrik is Assistant Professor in Product Development at the Department of Mechanical Engineering, Jönköping University. He teaches and researches in product development and computer supported engineering design. Recent research activities are directed towards engineer-to-order processes, design for manufacture, design automation and knowledge management.