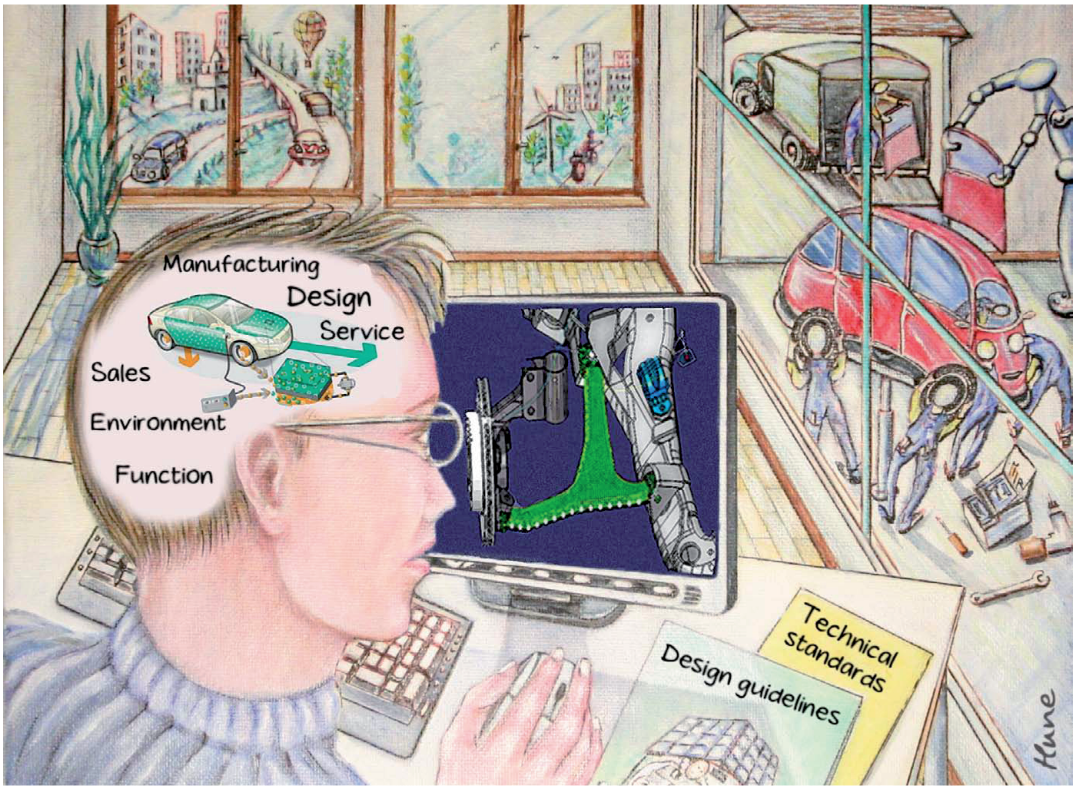
CHALMERS



Knowledge-based Engineering in Product Development Processes

Process, IT and Knowledge Management perspectives

AMER ĆATIĆ

Department of Product and Production Development
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2011

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Knowledge-based Engineering in Product Development Processes

Process, IT and Knowledge Management perspectives

AMER ĆATIĆ

Department of Product and Production Development
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2011

Knowledge-based Engineering in Product Development Processes

Process, IT and Knowledge Management perspectives

AMER ĆATIĆ

ISBN 978-91-7385-575-4

Doktorsavhandlingar vid Chalmers tekniska högskola

Ny serie nr 3256

ISSN 0346-718X

Published and distributed by:

Chalmers University of Technology

Department of Product and Production Development

Division of Product Development

SE – 412 96 Göteborg, Sweden

Telephone: +46 31 772 1000

URL: www.chalmers.se

Printed in Sweden by:

Chalmers Reproservice

Gothenburg, 2011

to Alma, Benjamin and Almina with love

"More gold has been mined from the thoughts of men than has been taken from the earth."

[Napoleon Hill]

Abstract

Product development as a field of practice and research has significantly changed due to the general trends of globalization changing the enterprise landscapes in which products are realized. The access to partners and suppliers with high technological specialization has also led to an increased specialization of original equipment manufacturers (OEMs). Furthermore, the products are becoming increasingly complex with a high functional and technological content and many variants. Combined with shorter lifecycles which require reuse of technologies and solutions, this has resulted in an overall increased knowledge intensity which necessitates a more explicit approach towards knowledge and knowledge management in product development.

In parallel, methods and IT tools for managing knowledge have been developed and are more accessible and usable today. One such approach is knowledge-based engineering (KBE), a term that was coined in the mid-1980s as a label for applications which automate the design of rule-driven geometries. In this thesis the term KBE embraces the capture and application of engineering knowledge to automate engineering tasks, regardless of domain of application, and the thesis aims at contributing to a wider utilization of KBE in product development (PD). The thesis focuses on two perspectives of KBE; as a process improvement IT method and as a knowledge management (KM) method. In the first perspective, the lack of explicit regard for the constraints of the product lifecycle management (PLM) architecture, which governs the interaction of processes and IT in PD, has been identified to negatively affect the utilization of KBE in PD processes. In the second perspective, KM theories and models can complement existing methods for identifying potential for KBE applications.

Regarding the first perspective, it is concluded that explicit regard for the PLM architecture decreases the need to develop and maintain software code related to hard coded redundant data and functions in the KBE application. The concept of service oriented architecture (SOA) has been found to enable an the explicit regard for the PLM architecture.. Regarding the second perspective, it is concluded that potential for KBE applications is indicated by: **1.)** application of certain types of knowledge in PD processes **2.)** high maturity and formalization of the applied knowledge **3.)** a codification strategy for KM and **4.)** an agreement and transparency regarding how the knowledge is applied, captured and transferred.

It is also concluded that the formulation of explicit KM strategies in PD should be guided by knowledge application and its relation to strategic objectives focusing on types of knowledge, their role in the PD process and the methods and tools for their application. These, in turn, affect the methods and tools deployed for knowledge capture in order for it to integrate with the processes of knowledge origin. Finally, roles and processes for knowledge transfer have to be transparent to assure the motivation of individuals to engage in the KM strategy.

Keywords: Knowledge based engineering, product development, product lifecycle management, knowledge management, service oriented architecture

Acknowledgements

To begin with I would like to gratefully acknowledge the FFI programme at the Swedish Governmental Agency for Innovation Systems (VINNOVA) that has financially supported this work which has been carried out at the Department of Product and Production Development at Chalmers University of Technology in a joint venture with a large global automotive company (referred to as Global Corp.).

A great deal of gratitude is directed to my main supervisor Professor Johan Malmqvist for his commitment and support during this work, helping me understand the issues as well as guiding me in doing research in the field of product development.

In addition MSc Sara Woxneryd at Global Corp. is acknowledged for being a supportive and helping companion during the two research projects. A very special thank you goes to Dr Hans Persson at Global Corp. for his support, commitment, positive attitude and encouragements which have significantly helped the research project in general and me personally in particular.

I would like to thank my colleague Dr Dag Bergsjö for a great collaboration and for being a source of many good ideas, initiatives and discussions. I hope we will continue our great collaboration for the years to come!

Another special thank you goes to Dr Trond Zimmerman, Dr Petter Andersson, Dr Ulf Högman and Professor Ola Isaksson at Global Corp. for the collaboration, for sharing many great insights and for showing what “The Global Corp. way” is all about.

I would like to express my gratitude to Professor Michael Vielhaber from University of Saarland for the collaborations and the perspectives on product development in general. I hope we will continue to collaborate around new ideas and concepts in the various topics of product development research.

A special thank you is directed to MSc Frank Müller at Mercedes-Benz Cars R&D who provided me with the possibility to spend some time as a guest researcher and work on interesting topics in a different setting. Another thank you to Dr Zbigniew Ragan who was an interesting and helpful companion in this endeavour.

Furthermore, I want to greatly acknowledge my colleagues at the department for making the everyday work enriched with discussions about everything other than KBE.

A very special place here and a very, very special place in my life is reserved to my wife Almina Ćatić together with our beloved daughter and son Alma and Benjamin. I want to thank the three of you for the endless support, encouragement, inspiration and the loving smiles you give me every day. I love you, mačaks!

I would also like to thank my parents, Suada and Zahid Ćatić, who have always encouraged me in life, and my brother Ermin Ćatić, for being a friend and a companion ever since we were kids. Finally, a special thank you goes to my father-in-law, Husnija “Hune” Kriještorac, for showing great interest in my work and for painting the image that decorates the front of this thesis with great dedication and enthusiasm.

Göteborg, Sweden, August 2011

Amer Ćatić

Appended papers and distribution of work

Paper A

Ćatić, A., and Malmqvist J. (2007) *Towards Integration of KBE and PLM* Proceedings of ICED07: International Conference on Engineering Design, Paper no. 540, Paris, France

The interviews were prepared and carried out by Amer Ćatić. The paper was written by Amer Ćatić. Johan Malmqvist contributed as a reviewer.

Paper B

Bergsjö, D., **Ćatić, A.**, and Malmqvist J. (2008) *Implementing a Service Oriented PLM Architecture Focusing on Support for Engineering Change Management* International Journal of Product Lifecycle Management, Vol. 3, No. 4, pp 335-355

The implementation of the demonstrators was initiated by Amer Ćatić. The programming effort was carried out as part of a master thesis project that Dag Bergsjö and Amer Ćatić supervised together. The paper was written by Dag Bergsjö and Amer Ćatić together. Johan Malmqvist contributed as a reviewer.

Paper C

Ćatić, A., and Andersson P. (2008) *Manufacturing Experience in a Design Context Enabled By a Service Oriented PLM Architecture* Proceedings of DETC08: ASME Design Engineering Technical Conferences – Design for Manufacturing and the Lifecycle Conference, Paper ID DETC2008-49858, 3 – 6 August, New York City, NY, USA

The idea for the demonstrator was initiated by Petter Andersson. The interviews and work with demonstrator concepts were carried out by Amer Ćatić and Petter Andersson together. The paper was written by Amer Ćatić and Petter Andersson together.

Paper D

Ćatić, A., and Malmqvist J. (2010) *Requirements Management When Introducing New Mechatronic Sub-Systems - Managing The Knowledge Gaps*, Proceedings of 11th International Design Conference - DESIGN 2010, Dubrovnik, Croatia pp. 661-672

The case study was planned by Amer Ćatić. The interviews were carried out by Amer Ćatić with assistance from Sara Woxneryd at Global Corp. The data was analysed by Amer Ćatić. The paper was written by Amer Ćatić. Johan Malmqvist contributed as a reviewer.

Paper E

Ćatić, A., and Malmqvist J. (2011) *Knowledge management in mechatronic product development: effective method for creating engineering checklists*, Submitted for journal publication

The method was developed and implemented in the macro application by Amer Ćatić. The evaluation was carried out by Amer Ćatić. The paper was written by Amer Ćatić. Johan Malmqvist contributed as a reviewer.

Other publications

Bergsjö, D., **Ćatić, A**., and Malmqvist J. (2008) *Implementing a Service Oriented PLM Architecture Using PLM Services 2.0* Proceedings of DESIGN 2008, pages 271 – 280, 19 – 22 May, Dubrovnik, Croatia

Ćatić, A., Bergsjö, D. and Malmqvist J. (2008) *Supporting Engineering Change Management by Integrating KBE and PLM in a Service Oriented Architecture* Proceedings of PLM08, Paper ID 216, 9 – 11 July, Seoul, Korea

Bergsjö, D., **Ćatić, A**., and Malmqvist J. (2008) *Towards Integrated Modelling of Product Lifecycle Management Information and Processes* Proceedings of NordDesign 2008, pages 253 – 264, 21 – 23 August, Tallinn, Estonia

Zimmerman, T., Malmqvist J., and **Ćatić, A.** (2008) *Implementing PLM In Practice – Findings From An Empirical Study In the Commercial Vehicle Industry* Submitted to Research in Engineering Design

Ćatić, A., Vielhaber, M., Bergsjö, D., Bitzer, M. (2009) *Automotive PLM - Applying a Product Design Mindset in PLM Implementation,* Proceedings of 2nd Nordic Conference on Product Lifecycle Management - NordPLM '09, Gothenburg, Sweden

Ćatić, A. (2009) *Integrating KBE and PLM Through a Service Oriented Architecture*, Licentiate thesis, Chalmers University of Technology, Göteborg, Sweden

Kero, T., Bergsjö, D., **Ćatić, A.**, Malmqvist, J., Söderberg, R., Andersson, M. (2009) *Assessing Methods for Management of R&D Process Information*, Proceedings of 18th *International Conference on Management of Technology, IAMOT 2009, Orlando, Florida, USA*

Vielhaber, M., Bergsjö, D. and **Ćatić, A.** (2010) *Mechatronic Systems Engineering - Theory and Automotive Practice*, Proceedings of 11th International Design Conference - DESIGN 2010, Dubrovnik, Croatia pp. 975-984

Ćatić, A. and Malmqvist, J. (2010) *Implementing a Wiki to Capture and Share Engineering Knowledge*, Proceedings of NordDesign 2010, Gothenburg, Sweden pp. 171-184

Ćatić, A. and Vielhaber, M. (2011) Lean Product Development: Hype or Sustainable New Paradigm?, Proceedings of 18th International Conference on Engineering Design 2011 (ICED 11), Copenhagen, Denmark, Vol. 1, pp. 157-168.

Table of contents

1 INTRODUCTION ... 1
1.1 Knowledge management in product development ... 2
1.2 Knowledge-based Engineering (KBE) ... 3
1.3 KBE in product development ... 4
1.3.1 Issues with KBE from a process and IT perspective ... 5
1.3.2 Issues with KBE from a knowledge management perspective ... 5
1.4 Purpose and goals ... 7
1.5 Delimitations ... 7
1.6 Outline of the thesis ... 8
2 FRAME OF REFERENCE ... 11
2.1 Knowledge-based Engineering (KBE) definitions ... 11
2.1.1 KBE development methodologies ... 14
2.1.2 KBE applications ... 20
2.1.3 KBE development tools ... 22
2.1.4 Conclusions about KBE applications and development methodologies ... 23
2.2 Knowledge management in product development ... 24
2.2.1 Knowledge in product development ... 25
2.2.2 Knowledge management approaches ... 27
2.2.3 Knowledge transfer, roles and motivation ... 29
2.2.4 Conclusions regarding knowledge management in product development ... 32
2.3 Product development processes ... 33
2.3.1 New Product Development ... 33
2.3.2 Lean Product Development ... 35
2.3.3 Configuration Design ... 38
2.3.4 Engineering Change Management ... 39
2.3.5 Conclusions about product development processes ... 40
2.4 Business Process Redesign ... 41
2.4.1 Methods for Process Improvement ... 41

2.4.2 IT and Process Improvement ... 43
2.4.3 Conclusions regarding business process redesign ... 44
2.5 Product Lifecycle Management (PLM) ... 45
2.5.1 Enterprise architecture ... 46
2.5.2 PLM architecture ... 46
2.5.3 KBE and PLM ... 47
2.5.4 Service-oriented architecture (SOA) ... 48
2.5.5 Conclusions regarding product lifecycle management ... 49
2.6 Conclusions and identified research gaps ... 50
3 SCIENTIFIC APPROACH ... 53
3.1 Research questions ... 53
3.2 Research model and empirical setting ... 54
3.2.1 Research model ... 55
3.2.2 Research setup ... 55
3.2.3 Research process and design ... 56
3.2.4 Validation approach ... 59
3.3 Adopted research approach ... 60
3.3.1 Review of KBE literature ... 60
3.3.2 Paper A – Towards integration of KBE and PLM ... 61
3.3.3 Paper B – Implementing a service-oriented PLM architecture focusing on support for engineering change management ... 62
3.3.4 Paper C – Manufacturing experience in a design context enabled by a service-oriented PLM architecture ... 64
3.3.5 Paper D – Requirements management when introducing new mechatronic subsystems – managing the knowledge gaps ... 64
3.3.6 Paper E – Knowledge management in mechatronic product development: effective method for creating engineering checklists ... 67
3.3.7 Overall validation approach in the research project ... 68
3.4 Summary ... 68
4 RESULTS ... 69
4.1 Paper A: Towards integration of KBE and PLM ... 69

4.2 Paper B: Implementing a service-oriented PLM architecture focusing on support for engineering change management ... 70

4.3 Paper C: Manufacturing experience in a design context enabled by a service-oriented PLM architecture ... 71

4.4 Paper D: Requirements management when introducing new mechatronic subsystems – managing the knowledge gaps ... 73

4.5 Paper E: Knowledge management in mechatronic product development: effective method for creating engineering checklists ... 74

4.6 Summary ... 76

5 FRAMEWORK FOR FORMULATING EXPLICIT KNOWLEDGE MANAGEMENT STRATEGIES IN PRODUCT DEVELOPMENT ... 77

5.1 Purpose and design of EFFEKT ... 77

5.1.1 EFFEKT from a methodological perspective ... 78

5.1.2 EFFEKT from a knowledge management framework perspective ... 81

5.2 Adopted perspective on knowledge management in EFFEKT ... 82

5.3 EFFEKT elements ... 84

5.3.1 Exploitation ... 84

5.3.2 Exploration ... 85

5.4 EFFEKT process ... 86

5.4.1 As-is knowledge application profile documentation ... 87

5.4.2 To-be knowledge application profile ... 91

5.4.3 To-be knowledge capture profile ... 92

5.4.4 Knowledge development (EFFEKT element 3-E and 3-F) ... 94

5.5 Identifying potential for KBE with EFFEKT ... 95

5.6 Summary ... 98

6 DISCUSSION ... 99

6.1 Discussion of the research questions and results ... 99

6.2 Discussion of EFFEKT from a knowledge management perspective ... 103

6.2.1 Knowledge transfer aspects in EFFEKT ... 103

6.2.2 Motivational aspects in EFFEKT ... 104

6.3 Discussion of the contribution ... 105

6.4 Discussion of the research setup and process ... 106

6.5 Summary ... 107
7 CONCLUSIONS ... 109
8 FUTURE WORK ... 111
REFERENCES ... 113
APPENDIX A
PAPER A
PAPER B
PAPER C
PAPER D
PAPER E

"An investment in knowledge pays the best interest."
[Benjamin Franklin]

1 Introduction

The field of product development, both in industry and in academia, is focused on three main business goals: shorter lead times, and lower product costs, with sustained or even increased product quality. These business goals are widely addressed across the product development research community which aims to develop technologies, methods, processes and tools to support product development.

Product development as a field of practice and research has significantly changed due to the general trends of globalization. Specifically, these trends have affected product development by changing the enterprise landscapes in which products are realized. Product development is performed with a variety of partners and sub-suppliers from different countries. The access to partners and suppliers with high technological specialization has also led to an increased specialization of original equipment manufacturers (OEMs). The OEMs used to have all those specialties in-house, but now concentrate only on specialties which make them unique and are relevant for their core business. The generally increased specialization has resulted in an overall increased "knowledge intensity" among both OEMs and suppliers, making knowledge an asset in a similar manner as equipment (Dunning, 2002).

In parallel to this development, a technological development in the field of support tools for product development has occurred. The development of information technology has led to advanced computer-aided design (CAD) tools for creating and managing geometric and physical properties, computer-aided engineering (CAE) tools for performing different kinds of analyses, and product data managing (PDM) tools which provide a basis for managing different kinds of product information in a variety of complex product structures and relate it to the product development process under the more abstract umbrella of product lifecycle management (PLM). Simultaneously with these systems' abilities to create and manage information, their abilities to accommodate and manage different kinds of engineering knowledge have increased.

In the knowledge-intensive business context, managing critical knowledge can mean the difference between market success and failure. There is a need to capture, reuse and maintain knowledge related to the product and related to the process for developing the product. Historically, when products were developed and manufactured by the same person, knowledge resided in the minds of craftsmen and its management was implicit. In the subsequent industrial revolution products and processes became more complex and the knowledge related to them became dispersed across many individuals. Combined with a higher utilization of consultants, increased involvement of partners and suppliers and a higher turnover of personnel, it is necessary for each organization to formulate an explicit approach for managing its unique knowledge to enable a continuous development of products and processes and secure the long-term survival of the respective companies. Among other methods, and following the trail of new computerized engineering tools, new technologies aiming to capture and reuse product and process knowledge to automate engineering tasks have been developed. This area is here referred to as knowledge-based engineering (KBE).

1.1 Knowledge management in product development

Product development is widely accepted as a knowledge-intensive process (Ullman, 2003; McMahon et al., 2004; Ward, 2007; Kennedy et al., 2008) where knowledge is created, shared and applied to develop new and innovative products and services. The agenda for knowledge management in product development has, however, been technologically driven by IT tools with new abilities, in the form of e.g. KBE, with focus on issues related to software application (Pinfold and Chapman, 1999; Penoyer et al., 2000; Stokes, 2001; Poenisch and Clark, 2006; Bermell-Garcia and Fan, 2008) rather than process utilization and knowledge management. A few contributions have reflected upon how knowledge management strategies can be applied to the context of product development (McMahon et al., 2004; Fu et al., 2006; Revilla et al., 2010). Another class of contributions has specifically focused on developing ontologies aimed at prescribing knowledge structures in the context of product development based on both theoretical and empirical studies (Ahmed, 2005; Ahmed et al., 2005).

The increased consciousness of the knowledge-intensive economy has brought about a higher awareness of the need for a more explicit approach to knowledge management in product development. The example of Toyota, documented in the Lean Product Development literature (Morgan and Liker, 2006; Ward, 2007; Kennedy et al., 2008), is stating a case in point which clearly demonstrates an explicitly formulated knowledge management strategy supported with roles, processes, methods and tools that has had positive effects on product development in terms of both product cost, lead times and product quality and innovation.

The notion of ‘knowledge management’ has been criticized from a philosophical and epistemological perspective (Styhre, 2003) due to the ambiguity of both the terms ‘knowledge’ and ‘management’. From the perspective of this thesis the notions of the terms ‘knowledge’ and ‘knowledge management’ are strongly related to the context of product development, which provides a frame for what can be considered as ‘knowledge’ and subsequently also what can be considered as the management of that knowledge. This issue is elaborated in more detail in Section 2.2.

1.2 Knowledge-based Engineering (KBE)

The coining of the term knowledge-based engineering (KBE) is associated with the launch of the rule-based CAD software ICAD in 1984 (Rosenfeld, 1995). The CAD-integrated applications of KBE have led to the perspective on KBE as a CAD technology rather than as an engineering method (Pinfold and Chapman, 1999). Blount and Kneebone (1995) argue however that the history of KBE is closely related to the roots of artificial intelligence applied to engineering problems, thus reflecting the perspective of KBE as a methodology rather than a particular commercial software. From their viewpoint the area as such has thus existed as long as digital engineering tools have existed, i.e. since the mid-1960s. Penoyer et al. (2000) share a similar perspective and focus on the role of KBE in the product development process, viewing it as an engineering method whose purpose is to automate engineering tasks through application of engineering knowledge.

There is still no universally accepted definition of exactly what is embraced by the term KBE. There are several proposed definitions which are more closely examined in Section 2.1. The following definitions are inspired by and derived from those existing definitions. They constitute my contribution to this debate and are used in this work:

> ***Knowledge-based Engineering (KBE)*** *is a strategic knowledge management method applying explicit engineering knowledge and IT solutions to automate engineering tasks. (Pinfold and Chapman, 1999; Whitney et al., 1999; Penoyer et al., 2000; Stokes, 2001; Poenisch and Clark, 2006)*
>
> ***KBE applications*** *are IT applications applying product and process knowledge in a computer-executable form to automate one or several engineering tasks. (Whitney et al., 1999; Schreiber et al., 2000; Stokes, 2001; Poenisch and Clark, 2006)*
>
> ***KBE development tools*** *are IT tools whose purpose is to support the creation of KBE applications. (Whitney et al., 1999; Schreiber et al., 2000; Stokes, 2001)*

These definitions reflect the perspective of KBE as a methodology that is embodied in IT solutions (without delimiting it to a specific kind of IT) with the main purpose to automate. The consequent use of the term “engineering tasks” implies that KBE supports several engineering domains such as mechanics, electronics and software engineering, as well as engineering functions dealing with the different lifecycle phases of the product – planning, design, manufacturing, aftermarket etc. In turn, this implies that KBE can be integrated in any area of IT supporting the activities in the respective field where product and process knowledge can be captured, encoded and executed to automate tasks. Thus KBE does not belong to any specific area of product development IT systems such as CAD, CAM, CAE, PDM etc.

The lack of a universally accepted definition of KBE also makes it hard to position the term exactly in relation to other kinds of so called knowledge intensive systems such as design automation, knowledge-based systems and expert systems. Figure 1-1 schematically illustrates the perspective on KBE in relation to the other types of systems adopted in this thesis. The highlighted area indicates that KBE largely overlaps design automation with more similarities than differences of significance for this thesis. The large overlap is mainly due to the inclusion of the terms automation and knowledge, as illustrated by the following definition of design automation as:

> *Engineering IT support by implementation of information and knowledge in solutions, tools, or systems that are pre-planned for reuse and support the progress of the design process. The scope of the definition encompasses computerised automation of tasks that directly or indirectly are related to the design process in the range of individual components to complete products (Elgh and Cederfeldt, 2005).*

The main difference between the two is reflected in the fact that design automation is automation enabled by knowledge while KBE also entails the ambition to manage the knowledge. The areas of knowledge-based system (KBS) and expert systems (ES) are here mainly associated with the use of a specific software architecture including an inference engine that executes a knowledge base. These are considered to overlap with KBE in case they are used for engineering purposes such as product configuration, while they underlap with KBE when used for other purposes such as e.g. medical diagnosis.

1.3 KBE in product development

From the perspective of this thesis it is interesting to highlight the fact that KBE still has not experienced widespread utilization in product development practice. Regardless of whether one defines KBE to have come into existence in the mid 1960s or mid 1980s its impact on product development has still not reached the same level as that of other digital engineering tools and methods. The reasons for this have been claimed to be several. Bachrach (1997) found that a key factor for successfully identifying potential for design automation applications consists of people who are knowledgeable enough in both the product development process and the abilities of design automation technology. And these people are scarce in any product-developing organization. Another area that is considered as a barrier for why KBE has not experienced widespread use is the lack of interoperability between the different IT systems in which KBE can

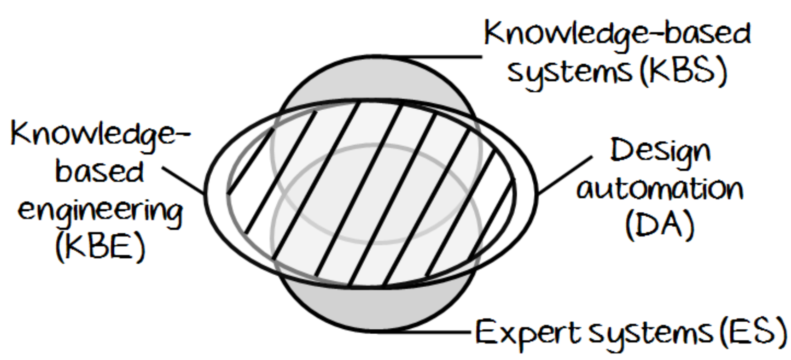


Figure 1-1: The perspective on how KBE relates to other kinds of knowledge intensive systems

be implemented (OMG, 2005). This issue has been highlighted and described in more detail, but without any concrete proposals for how it should be addressed (Penoyer et al., 2000; Poenisch and Clark, 2006; Bermell-Garcia and Fan, 2008). Literature regarding KBE in product development practice is scarce. Some examples of companies, such as Airbus, exist which have developed KBE strategies and implemented KBE in a structured manner in terms of not only new tools but also methods and processes to capture and reuse engineering knowledge (Haas and Sinha, 2004). Otherwise it is more a question of successfully implemented demonstrators, such as (Kochan, 1999; Merkel and Schumacher, 2003; Boart et al., 2006) than operational methods supported with IT applications rolled-out in production such as (Strinning, 1995; Fuxin, 2005).

1.3.1 Issues with KBE from a process and IT perspective

The literature regarding KBE applications sheds little light on issues regarding the integration of KBE with processes and IT. The publications deal with demonstrating the abilities of KBE in different contexts of product development such as automation of design (Cederfeldt, 2007), configuration (Fuxin, 2005), analysis (Merkel and Schumacher, 2003; Boart et al., 2006) and information processing tasks (La Rocca et al., 2002) when the product definition flows between e.g. design and different kinds of analysis.

When viewed strictly from a process and IT perspective, an issue labeled “isolated islands” has been identified with KBE in product development. The label denotes an “isolation” of the KBE applications from *both* the processes and IT (OMG, 2005). From a process point of view, this means that only a limited number of activities are supported (compared to how many could potentially be supported) in the product development process, and has implications for the business operations and goals. From an IT point of view, the isolation means that the applications are detached from the IT environment which has implications for processes (such as the need to manually take care of the KBE applications inputs, outputs, updating etc.) as well as for how KBE is implemented in terms of abilities to realize the full potential of process support once a possible KBE application is identified. The previously mentioned “successful demonstrators” typically depict such isolated islands, even if they are implemented in an industrial process. The issue of “isolated islands” is focused upon in this thesis and is largely considered to be a symptom of a lack of explicit regard for the constraints of the PLM architecture in the development of KBE applications, detailed in Section 1.4 which states the purposes and goals of the thesis.

1.3.2 Issues with KBE from a knowledge management perspective

The state of practice for KBE implementation reveals a lack of methods for identifying where KBE applications are feasible (Blount and Kneebone, 1995; Bachrach, 1997). A rare exception is Sunnersjö (1994) who devised a chart in which characteristics of the product and the product development process are used to indicate potential for automation of design; see Figure 1-2. No contributions have however been found that provide support for identifying potential for KBE

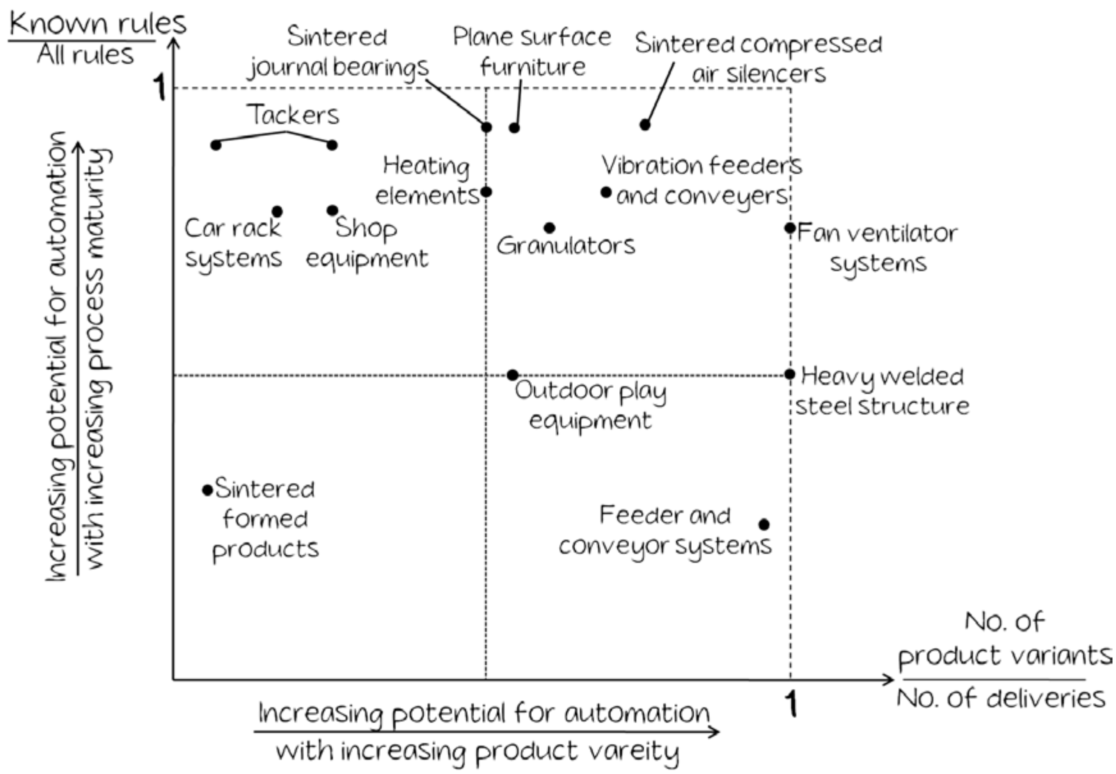


Figure 1-2: Identifying economic potential (x-axis) and process potential (y-axis) for design automation (Sunnersjö, 1994)

seen from a knowledge management perspective. By adopting this perspective, potential for KBE applications could be identified by considering whether and how KBE complies with the knowledge management strategy in a certain organization, which in turn could contribute to a wider utilization of KBE in product development processes. The lack of knowledge management perspectives in KBE development and implementation can be considered as a manifestation of the previously mentioned lack of explicit knowledge management perspectives in product development in general.

In the studied literature describing KBE applications (compiled in Appendix A) two areas dominate: the IT solution and the process improvement which justifies the IT solution. In this sense KBE resembles any other implementation of IT, with the distinguishing feature that it is based on, and contains, engineering knowledge. This is mentioned in all of the contributions listed in Appendix A, but seldom taken a step further and analyzed in terms of what is implied by the fact that KBE, in some sense, also manages knowledge (or is at least dependent on it). Contributions 11, 13, 15, 16 and 18 in Appendix A dedicate some paragraphs to this issue but are far from a thorough analysis. This issue is to a limited extent addressed by methodological frameworks (described in more detail in Section 2.1.1) but mainly in terms of the implications this has for the IT solution, which is in focus there.

1.4 Purpose and goals

The general purpose of this thesis is to propose and evaluate solutions to achieve a wider utilization of KBE in product development processes. This general purpose has been approached by adopting the two different perspectives of KBE as described in the previous subsection.

From the process and IT perspective, the aim is to describe the reasons behind the phenomenon of isolated islands, and to propose and evaluate solutions for addressing those reasons, with the purpose of contributing to a wider utilization of KBE in product development. To address this purpose and aim, the following research goal was formulated:

Research Goal A: Identify and evaluate concepts and solutions which can complement the current methods for KBE development and support the developers in developing and implementing well-integrated and reusable KBE applications.

From the knowledge management perspective, the aim is to identify potential for KBE by describing attributes and characteristics of the way knowledge is managed in an organization that indicates potential for KBE. The purpose of this is to contribute to an increased ability to identify potential for KBE applications, which in turn contributes to a wider utilization of KBE in product development processes. To address this aim the following research goal was formulated:

Research Goal B: Clarify which attributes in explicitly formulated knowledge management strategies indicate potential for KBE.

The addressing of this research goal is, however, strongly related to the ability to formulate explicit knowledge management strategies in product development. Therefore an additional research goal is stated in order to increase this ability, with the aim of proposing methodological support for doing so. The third research goal reflects the lack of dedicated frameworks for formulating explicit knowledge management strategies in the context of product development:

Research Goal C: Propose a framework to support product development practitioners in formulating explicit knowledge management strategies in their processes/organizations.

In order to break down these research goals into research questions we need to relate them to existing literature. The literature is critically reviewed and summarized in Chapter 2 with focus on identifying and describing relevant contributions so as to find research gaps. The research goals, gaps and previous contributions combined constitute the foundation that motivates the research questions, whose exact formulation has been postponed to Section 3.1 in order for the reader to gain an understanding of how they relate to existing literature.

1.5 Delimitations

The main contributions of this thesis are related to the cross-functional and applied research field of product development or “design science” (Hubka and Eder, 1996). Thus, a delimitation is that there is no primary intention of contributing to the original fields which have been applied, such as computer science, software engineering etc.

One of the main delimitations of this research is that no attention is given to the inner workings of KBE applications such as different systems' abilities, programming languages, semantics, algorithms, problem-solving techniques, reasoning mechanisms etc. As the main focus is on the way KBE applications are utilized in the product development process, KBE applications themselves will be considered as "black boxes" and only the interactions with the process and IT environment are of interest.

The delimitation in relation to the area of knowledge management is that the term "knowledge" embraces knowledge of relevance for either the products or the processes that exist in a product-developing organization and can be classified as either product or process knowledge. In terms of organizational boundaries the organization of primary interest, i.e. the "subject" towards which a knowledge management initiative is directed, is the organization developing the product. Due to the project partnership setup, the industries of interest for this research have been mainly the automotive and aerospace industries.

1.6 Outline of the thesis

Chapter 1 has covered the background and the research goals of the project along with some of the basic concepts regarding knowledge and KBE. In addition the issues with KBE from a product development perspective were detailed.

In ***Chapter 2*** the frame of reference for this project is described, beginning with a detailed analysis of KBE as such in terms of definitions, applications and categorizations, followed by methodological frameworks that can be used for developing KBE applications. The chapter continues with a deeper look into knowledge management theories and models, and product development processes seen from a knowledge perspective, followed by a closer look at business process redesign. Finally the area of PLM architecture is described, as this constitutes the IT environment in which KBE applications act. The chapter is concluded with the identification of research gaps related to KBE development methodology from a knowledge management perspective and a process and IT perspective, respectively.

Chapter 3 is initiated with the formulation of the four research questions and how they relate to the research goals stated in Chapter 1 and research gaps and contributions described in Chapter 2. The chapter is continued with the research process and research design, followed by the adopted research approach. The chapter is rounded off with a description and discussion of the adopted validation approach.

Chapter 4 summarizes the results from the appended papers in a paper-by-paper fashion.

Chapter 5 is devoted to describing a framework for formulating explicit knowledge management strategies in product development.

Chapters 6 discusses the results from the perspectives of the research questions and contribution, and it also discusses the research process and setup.

Chapters 7 and 8 conclude this thesis and propose ideas for future work, respectively.

In ***Appendix A*** 24 KBE applications found in the literature have been analyzed and their addressing of topics relevant for this thesis are summarized.

Finally the five papers that constitute the basis of this thesis are appended at the end.

"I start where the last man left off."
[Thomas Edison]

2 Frame of reference

In the previous chapter two different perspectives on KBE were mentioned: a process and IT perspective and a knowledge management perspective. Using these perspectives on KBE, this chapter is structured according to Figure 2-1, in which relevant topics are depicted.

This chapter summarizes the relevant literature in the depicted topics in order to describe the state of the art in the field of KBE as well as in the related fields. The purpose is to identify a set of research gaps regarding KBE in product development (summarized in Section 2.6), which closely relate to the research goals stated in Section 1.4. The research goals and identified research gaps then provide the background for the research questions stated in Section 3.1.

This chapter starts with the current state of KBE in product development, with definitions and a summary of methods for developing KBE and applications of KBE found in the literature. The chapter is continued with sub-topics from knowledge management such as strategies and methods, roles and motivation and knowledge transfer which are reviewed from the perspective of product development. This is followed by a summary of product development process frameworks which are reviewed from a knowledge management perspective, and the chapter is rounded off with frameworks and methods from the areas of business process redesign (BPR) and product lifecycle management (PLM) reviewed from a KBE perspective.

2.1 Knowledge-based Engineering (KBE) definitions

As stated in Section 1.2, there is still no universally accepted definition of exactly what is embraced by the term knowledge-based engineering. There is a heavy focus towards software application rather than methods focusing on process support. An example of this is Poenisch and Clark (2006) who agree that no definition of KBE has found general acceptance yet, but they argue that an essential ingredient of KBE is a software application that processes some kind of engineering knowledge. Blount and Kneebone (1995) consider KBE to have roots in artificial intelligence and more concretely to be an application of knowledge-based systems (KBS) or expert systems (ES) in engineering design and manufacturing. Therefore the areas of KBS (Guida and Tasso, 1995) and ES (Sriram, 1997; Jackson, 1999), which have a longer record of publications compared to KBE, are used as reference points in some of the aspects that are discussed in this thesis. How these areas relate to KBE is schematically illustrated in Figure 1-1.

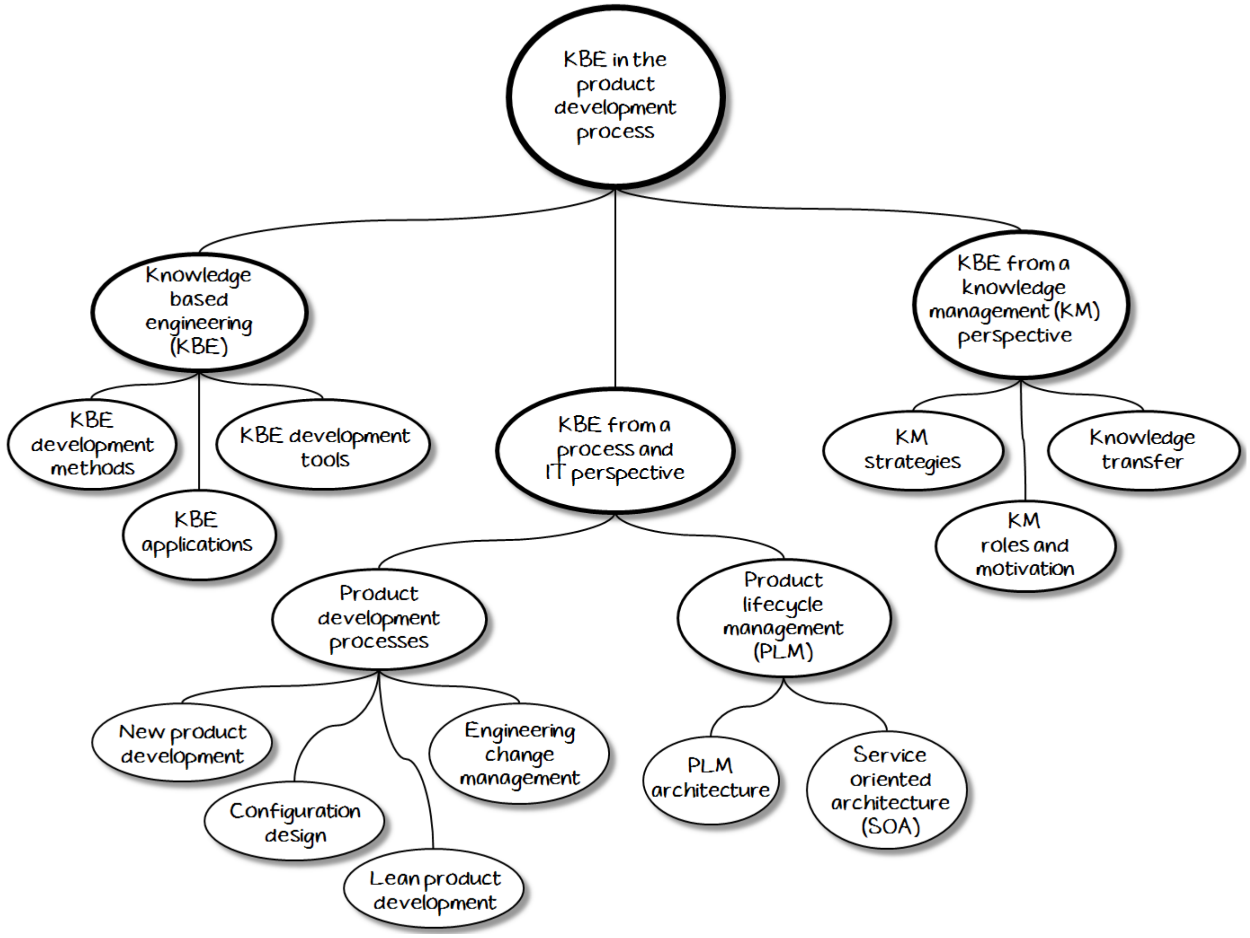


Figure 2-1: Overview of the research area

In the European project MOKA (Methodology and software tools Oriented to Knowledge Based Engineering Applications) (MOKA, 2000), KBE is defined as:

> *"The use of advanced software techniques to capture and reuse product and process knowledge in an integrated way". (Stokes, 2001)*

Penoyer et al. (2000) define KBE as computer systems used for engineering, focused on a representation and application of knowledge to specific problem cases. They state that the computer system should have deep penetration into the problem domain and reason through the problem-solving process using rules of logic rather than mathematical models, in order to distinguish it from computer-aided engineering (CAE).

Pinfold and Chapman (1999) go a step further and define KBE as a derivative of CAD where, besides geometry, also the design intent is captured – and therefore KBE is defined as a framework for capturing and defining the process of design creation. With this definition KBE is heavily connected to a certain kind of software (CAD systems) and related only to geometrical aspects of a product. This also corresponds to the way commercial CAD system vendors define the term, i.e. as CAD-integrated KBE modules such as CATIA's "Knowledgeware" (Dassault Systèmes, 2011) or NX's "Open Knowledge Automation" (Siemens PLM, 2011).

In the more general area of knowledge-based systems within the methodology of CommonKADS, there is no definition of KBE but the term "knowledge systems" is used as a gathering term for expert systems, knowledge-intensive information systems and knowledge-based systems (Schreiber et al., 2000). Guida and Tasso (1995) propose a methodology for design and development of knowledge-based systems, which considers the lifecycle of the knowledge-based system, called KLIC (Knowledge-based system LIfe Cycle). Their definition of a knowledge-based system is focused on the system's architecture consisting of a reasoning mechanism, a knowledge base and a working memory (whose purposes are purely operational). Guida and Tasso (1995) focus their definitions based on the characteristics which differentiate knowledge-based systems from traditional software systems, which are considered as purely procedural. A definition of expert systems is given by Jackson (1999) who states that:

> *"An expert system is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice"*

In addition, Jackson reasons around the architecture of an expert system similar to that of Guida and Tasso (1995) dividing it into a knowledge base and an inference engine (equivalent to the "reasoning mechanism" from Guida and Tasso (1995)) and with similar reasoning about the differences between an expert system and procedural software programs. For the sake of convenience the definition of KBE used in this thesis is repeated here:

> ***Knowledge-based Engineering (KBE)*** *is a strategic knowledge management method applying explicit engineering knowledge and IT solutions to automate engineering tasks. (Pinfold and Chapman, 1999; Whitney et al., 1999; Penoyer et al., 2000; Stokes, 2001; Poenisch and Clark, 2006)*
>
> ***KBE applications*** *are IT applications applying product and process knowledge in a computer-executable form to automate one or several engineering tasks. (Whitney et al., 1999; Schreiber et al., 2000; Stokes, 2001; Poenisch and Clark, 2006)*
>
> ***KBE development tools*** *are IT tools whose purpose is to support the creation of KBE applications. (Whitney et al., 1999; Schreiber et al., 2000; Stokes, 2001)*

These definitions are inspired by the previous definitions, the main distinction being related to issues regarding IT and software. The definition does not impose any delimitations to any particular subset of IT solutions, software types or techniques. The reason for this is that KBE is here primarily viewed as a kind of process improvement method enabled by explicit engineering knowledge, with the ultimate goal to automate administrative or repetitive tasks. The implication of the process improvement aspect is that any IT solution (or combination of solutions) with the potential to realize this goal is valid for consideration. The implication of the knowledge aspect is that the KBE application embodies the management of certain knowledge and, as such, has to be aligned with either an explicit or implicit knowledge management strategy in the organization where it is used. The keyword in the definition is "automation" which has to be feasible from both of the adopted perspectives.

2.1.1 *KBE development methodologies*

In the research project, four methodologies related to implementation of KBE have been analyzed. The first two methodologies are focused on implementation of knowledge-based systems, called KLIC and CommonKADS (Schreiber et al., 2000), which are positioned at a more general level supporting implementation of systems reusing any kind of knowledge (Guida and Tasso, 1995). The third methodology is MOKA (Stokes, 2001), whose articulated purpose is to support the implementation of KBE. The fourth methodology is proposed by Cederfeldt (2007), and is referred to as Cederfeldt's methodology. It has the articulated purpose of supporting the planning of design automation systems.

KLIC

The Knowledge-based system LIfe Cycle methodology (KLIC) was developed by Guida and Tasso (1995) in response to the need for a methodological framework for developing knowledge-based systems, leaving issues of specific techniques and technologies outside of the scope. The main phases of the KLIC methodology are depicted in Figure 2-3 and the aim is to cover the complete lifecycle of the KBS, in ways similar to how MOKA identifies the lifecycle of a KBE application. The main difference between the two frameworks is that KLIC also addresses both the early and the late phases in the lifecycle. In each of the phases a series of detailed tasks, supporting methods and defined outputs such as reports, lists, demonstrators and necessary decisions are described in detail from technological, organizational and project management perspectives. Given the scope of this thesis and the stated research goals, the phases of most interest are Phase 0 which identifies potential for a knowledge-based system and Phase 4 where the integration of the KBS with its surrounding IT environment comes into play.

In Phase 0, "opportunity analysis", the authors suggest a set of tasks which start with a definition of business objectives and goals and continue with analyzing the process and IT

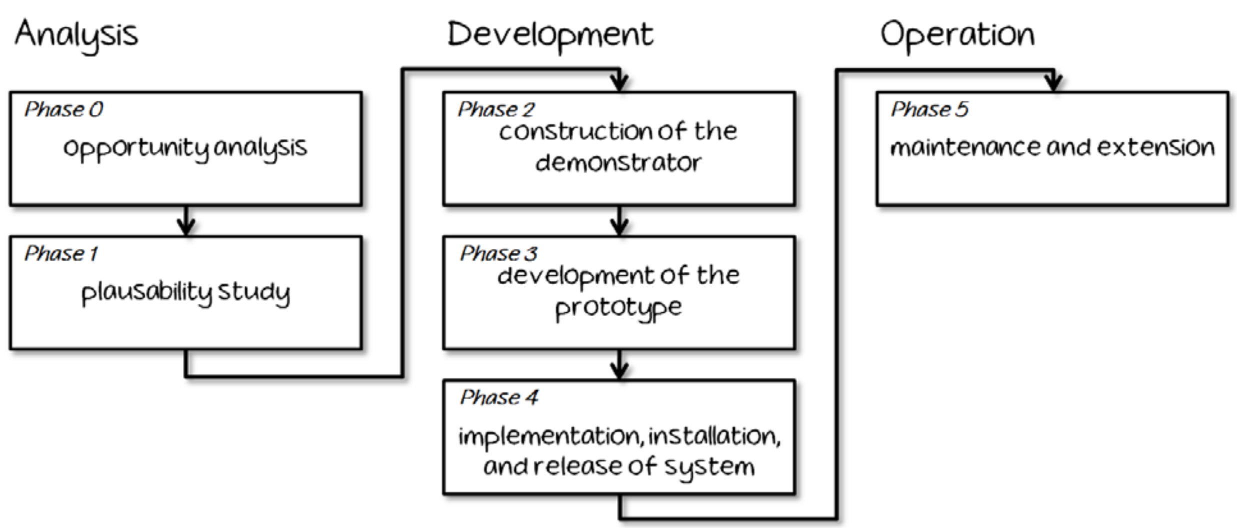


Figure 2-2: Main phases of the KLIC methodology (Guida and Tasso, 1995)

environment to identify and characterize areas and domains which are "knowledge-intensive". In addition to this, knowledge-related problems are defined in the three categories of knowledge losses, knowledge bottlenecks and knowledge waste, to be able to connect possible knowledge problems to a future KBS application. Finally opportunities for a KBS application are defined and characterized, where Guida and Tasso (1995) list aspects related to the technical appropriateness of the KBS application such as availability of knowledge sources, technical complexity and risk, and the usefulness for the organization such as the strategic fit and expected benefits. Guida and Tasso (1995) also provide a listing of "typical tasks" with high potential for KBS applications, such as configuration and diagnosis.

In respect to the process and IT perspective of KBE, Guida and Tasso (1995) have some points stated already in Phase 0 which are relevant for identifying potential for KBS applications, such as the definition of the overall business objectives, an analysis of the processes and IT layers (labeled "level of automation") and, in the end of Phase 0, integration with other software systems – which is highlighted as part of the analytical assessment of potential candidates along with the purpose of the KBS. In Phase 4, which deals with implementation, installation and release of the target system, IT integration is briefly mentioned, primarily as the need to analyze which interfaces the target system should have to surrounding systems and to prepare it for those.

CommonKADS

CommonKADS (Schreiber et al., 2000) is short for Common Knowledge Acquisition and Documentations Structuring and was developed as a response to the need for a standard for knowledge-based systems. Its development began in the early 1980s at the University of Amsterdam and was a synthesis of current methods used to develop expert systems processing different kinds of knowledge; engineering, diagnostics, planning and so on.

CommonKADS' view on KBS is similar to KLIC and MOKA dividing the knowledge into domain knowledge, inference knowledge and task knowledge. The distinction between inference knowledge and task knowledge is that the inference knowledge deals with different ways that the domain knowledge is executed, while task knowledge deals with which inference knowledge is applied depending on the goals and strategies for a particular task in a process. The method is essentially based on a set of models which describe a particular aspect needed or affected by the knowledge-based system. The model suite with relations is depicted in Figure 2-3.

The three models at the top reflect a mapping of the processes in the organization along with the roles and resources used in the execution of the processes, while the communication model is more related to IT aspects of how and which information is created, processed and communicated between different resources (agents). The knowledge model is more abstract and describes the knowledge that is used in the execution of the different tasks. Finally the design model constitutes the main input for the design of the KBS, where it is stated how the knowledge is structured along with how that knowledge acts to create and process the information specified by the communication model, and also how the KBS application interacts with its environment.

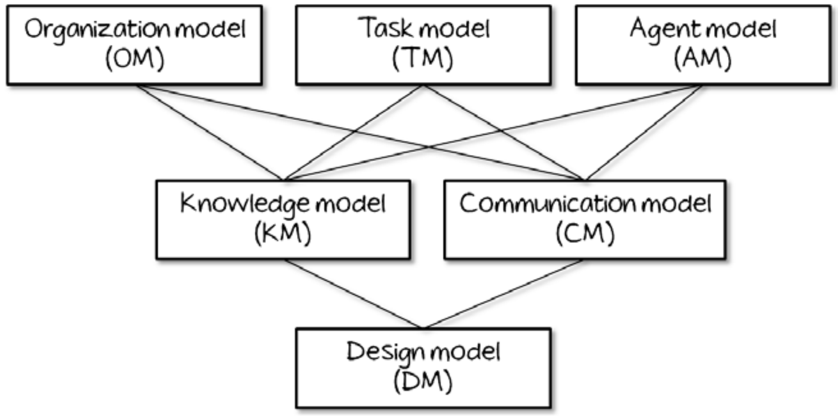


Figure 2-3: CommonKADS model suite and their relations (Schreiber et al., 2000)

Each of the models is made up of a set of worksheets labeled OM-1 – 4, TM-1 – 2 etc. (see Table 2-1 which illustrates OM-3 as an example) whose completion guides the process of specifying a KBS application.

Due to its aim to support the development of knowledge-based systems in general, the implementation of CommonKADS is very much dependent on the specific area of interest. Some support for the area of engineering is given in the knowledge models, where general templates are provided for configuration design (given a set of predefined components, constraints and requirements, find an assembly that satisfies requirements and obeys constraints) and synthesis (construction of a system structure which fulfills a given set of requirements).

The identification of potential KBS applications finds some support in the methods provided as part of the worksheets OM, TM and AM. These are general and based on ideas of mapping processes and reasoning about process problems and issues which are later assessed as potential cases for KBS implementation (similar to KLIC). The explicitly stated view of CommonKADS is that KBS is considered to be part of process improvement, which is also reflected in the initial steps.

Considering instead the way IT issues are addressed, the communication models make a substantial contribution to defining how the KBS is to interface its surrounding human and non-human agents. The generality of the approach, however, gives little support in the specific area of engineering and KBE. In addition to the communication models, Schreiber et al. (2000) also reason about the possibilities for a distributed architecture within the KBS itself. Some ideas that are elaborated are e.g. to separate the implementation of the inference engine as a "problem-solving" module or service that contains different problem-solving techniques and, depending on the task execution strategy, acts on any base of domain knowledge. This implies that the same inference engine could be used in different KBS applications.

Table 2-1: A process breakdown sheet for the template OM-3 (Schreiber et al., 2000)

Organisation model		Process breakdown sheet OM-3				
NO.	Task	Performed by	Where?	Knowledge asset	Intensive?	Significance
Task id.	Task name (from OM-2)	A certain agent, either a human (see People in OM-2) or a software system (see Resource in OM-2)	Some location in the organization structure (see OM-2)	List of knowledge resources used by this task	Boolean indicating whether the task is consdiered knowledge -intensive	Indication of how significant the task is considered to be (e.g. On a five point scale in terms of frequency, costs, resources or mission criticality)

MOKA

MOKA is an abbreviation of Methodology and tools Oriented to Knowledge-based engineering Applications (MOKA, 2000). It was developed in response to an increasing amount of available tools for developing KBE applications as well as an increasing number of implemented KBE applications in the industry. MOKA was a consortium consisting of European companies within the aerospace and automotive industries along with IT partners and one university.

MOKA defines the lifecycle of a KBE application as illustrated in Figure 2-2 with six different phases. The focus of the methodology itself is on the phases of capture and formalization of the knowledge. The reason for this is to provide support for documenting the knowledge (in “Capture”) and representing it in a format (in “Formalize”) that is independent of any commercial software. The two phases are supported with different tools which in turn provide complete traceability between the captured “raw” knowledge and the formalized computer executable knowledge. Capture is performed using a set of documents called ICARE, which is an abbreviation for:

Illustration – represent general knowledge with descriptions and comments.
Constraints – model interdependencies between entities in the product.
Activity – describe the various problem-solving steps in the design process.
Rules – describe the knowledge that controls the design process execution.
Entities – describe product object classes.

Once the knowledge is captured in the semi-formal format of ICARE documents with text and illustrations, it is then formalized using MOKA Modelling Language (MML). MML is based on UML (Unified Modelling Language, (OMG, 2008)) with a set of predefined classes, subclasses

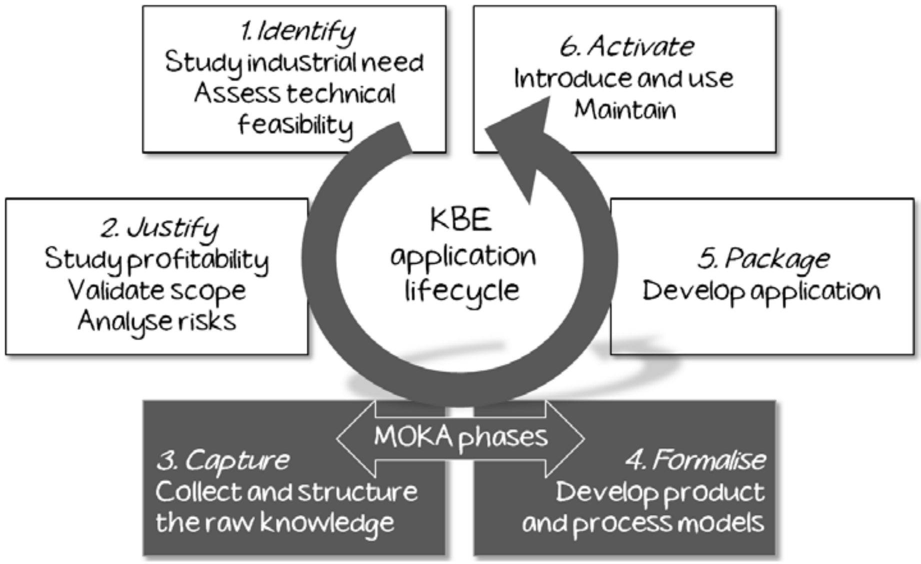


Figure 2-4: KBE applications lifecycle and focus of MOKA (Stokes, 2001)

and attributes which facilitate the formalization of knowledge and ensure full traceability between ICARE and MML. MML contains product-related classes, which are populated using the C and E documents, and process-related classes which are populated using the A and R documents. The I documents are reserved for illustrations and comments.

Essentially MOKA reflects the same application architecture for KBE as was described for knowledge-based systems, consisting of domain knowledge (which is structured using the product-related parts of MML) and problem-solving knowledge (which is structured using the process-related parts of MML and referred to as the inference engine in KBS literature).

As mentioned earlier, the initial steps of the KBE lifecycle (identification and justification) as well as the final steps of packaging and activation are not supported by the MOKA framework. Stokes (2001) states that the identification and justification of a KBE application does not principally differ from a KBS application, and refers to the KLIC methodology (Guida and Tasso, 1995) for more detailed support in those steps. In the final steps of packaging and activation, issues related to implementing the formal models into a KBE platform are briefly addressed as well as how to maintain existing KBE applications. In the activation step, issues regarding distribution and user acceptance are briefly described, once again with the explicit statement that these steps are not focused upon by the MOKA methodology.

Cederfeldt's methodology for planning design automation

The methodology provides guiding principles for the planning of design automation systems (Cederfeldt, 2007). In line with the reasoning and the definitions provided in Section 1.2, the term "design automation" is considered to be synonymous with KBE, especially when seen from a business process point of view. The main steps of Cederfeldt's methodology are described in Figure 2-5 and are divided into three different phases. In the first phase, the needs from a process point of view are addressed. The questions relate to the perceived need as stated by those working in the process, as well as the need imposed by the to-be situation of which kind of products the process wants to deal with. These aspects then relate to what kind of knowledge is to be implemented in the future application. Cederfeldt also presents a framework that supports the proposed methodology, in which certain constructs are defined and explained to guide the implementation of the methodology and provide support. Figure 1-2 is one of the referred constructs in which potential for design automation is schematically appreciated by considering the kind of product which the process to be supported deals with along with the process maturity, which according to Sunnersjö (1994) can relate to the appreciated percentage of known rules in relation to all rules which govern a design. Other identifiers of need are defined by outer constraints and requirements which relate the system to-be to different stakeholders and what they want the system to do. Other aspects considered in the first phase are those that contribute to the potential of the system to-be. Factors such as formalization of design knowledge and task, the process maturity and the appreciated return on investment all raise the potential of success for design automation.

In the second phase, the methodology provides support in mapping identified problem definitions to solution strategies. The solutions strategy is dependent on the task, the kind of knowledge, and the other requirements that delimit which solution principles can be applied.

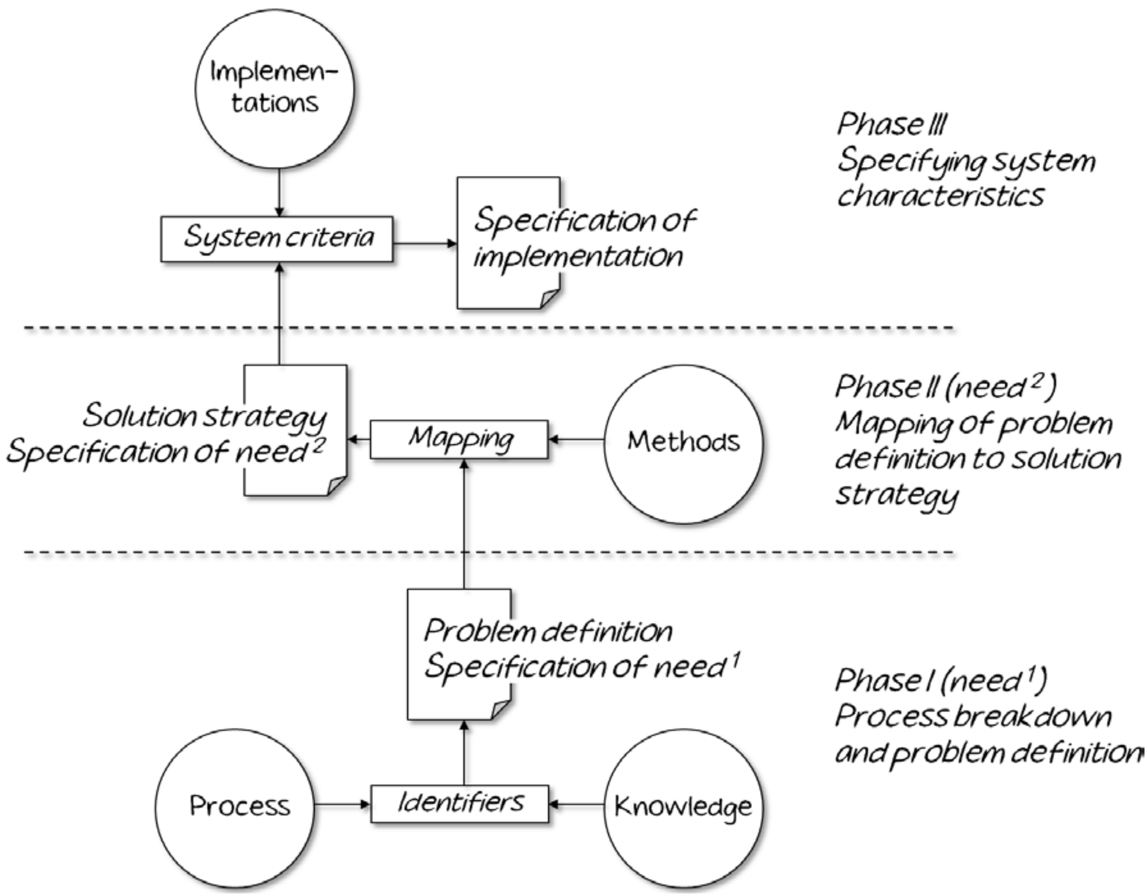


Figure 2-5: The main phases of the methodology for planning design automation (Cederfeldt, 2007)

In the third phase the system characteristics are specified in terms of transparency (clarity about what the application does and how), accessibility to the implemented knowledge, the flexibility, extensibility and longevity of the designed system, and its ease of use. The method also briefly discusses the system implementation in terms of the system's integration, cost, maintenance and education of the users.

In regard to the areas of interest for this thesis, identification of potential for KBE applications and process and IT considerations in KBE development, Cederfeldt explicitly states that the first one is out of scope for his methodology. The process and IT issues are briefly discussed in terms of a stated need to integrate the designed system with surrounding systems and people it needs to interact with during the task(s) it is supposed to perform.

2.1.2 KBE applications

In Appendix A, 24 KBE applications that have been found in the literature have been summarized, considering the aspects of how the need for the application is motivated, which purposes the application has, and whether its stated aim is process improvement or knowledge management. In addition, for each application it is summarized whether and how the application addresses issues related to IT architecture (primarily the PLM architecture), methodical approaches, and knowledge management.

As indicated in Figure 2-6 the applications exist in the following categories:

- Parametric design of rule-driven geometries
- Configuration
- Generation of models or preprocessing of models for frequent kinds of analysis of specific characteristics
- Simulation of specific properties with constraints and rules governing the simulation task.

Some cover several steps and some are focused on providing support in only one. The most common motivation for each application is that it automates a time-consuming and repetitive task for which a set of governing rules exists in the product domain and/or the process domain. The purpose, from an academic point of view, is usually to demonstrate some specific framework for automating the task, structuring and representing the knowledge or demonstrating specific tools.

Regarding the methodological approaches adopted for the development of the KBE applications, only a few of the authors provide any details. Two of the applications, CoRPP (Elgh and Cederfeldt, 2005) and DART (Pinfold and Chapman, 1999), provide transparency regarding their approaches, which are mostly related to software development in general. In the development of CoRPP (Elgh and Cederfeldt, 2005) the frameworks provided by Jackson (1999) and by Hopgood (2001) are referenced. These are, however, mostly focused on issues regarding implementing appropriate problems-solving techniques in the inference engine or choosing appropriate programming frameworks.

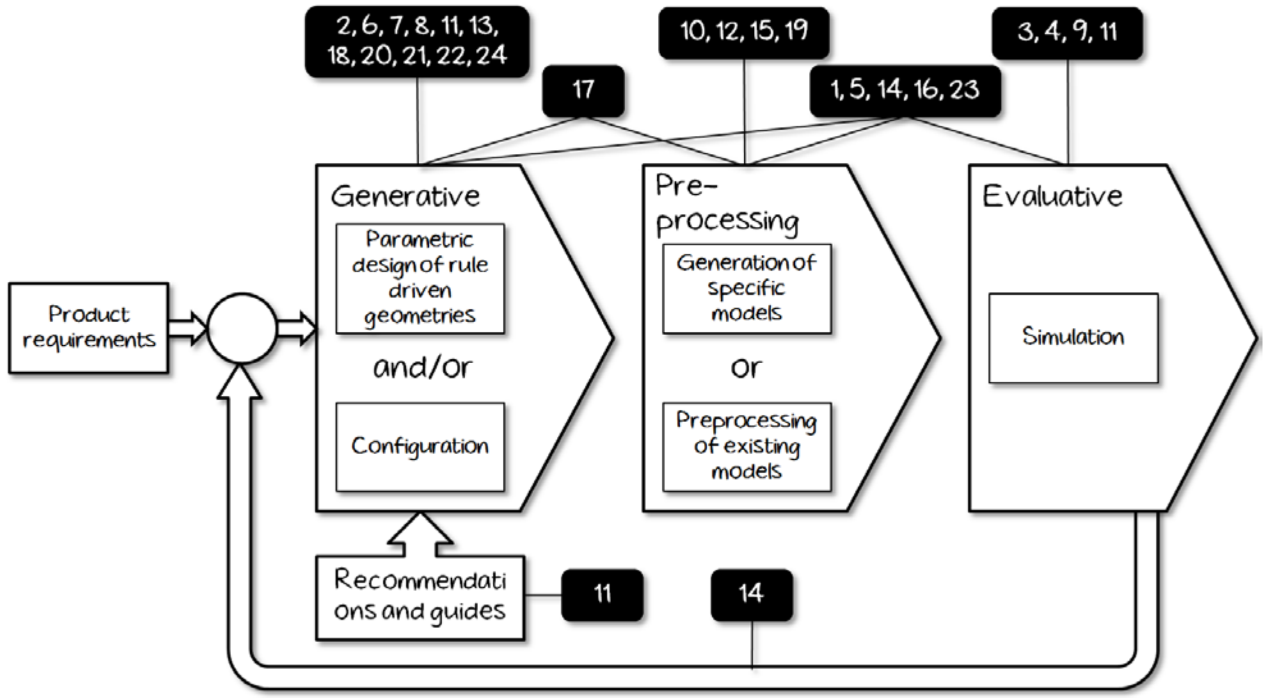


Figure 2-6: A schematic categorization of tasks performed by KBE applications summarized in Appendix A

Regarding the issue of how the potential for the KBE applications was identified, none of the published works found has presented any approaches. All authors present reasoning about the benefits of the application when used to support the intended task. Typically in the introductory sections a global challenge for product-developing companies, such as needs for shorter lead times or reduced development cost, is mentioned in order to motivate the existence of the application. Following the global challenges is reasoning about the time-consumption and the repetitiveness of the focused task, to explain why the chosen task is a good candidate for automation. There are no descriptions of how the authors scoped down a particular area such as manufacturing preparation, configuration design, redesign or finite element analysis and then identified a particular task.

Regarding the process and IT perspective, the image is somewhat different. The issue of the internal architecture of the KBE application is addressed by several of the authors. In most cases, commercial applications are used for performing different tasks of the total application. Their distribution and partial roles are clearly addressed and explained with the interfaces between them. Focusing instead on the position and role of the implemented KBE application in relation to its surrounding IT and processes, a few contributions provide this kind of considerations (numbers 6, 13 and 18). Some of the others, such as numbers 1, 8, 15 and 17, do consider IT aspects related mainly to the need to interface other applications and systems. The other contributions provide no considerations regarding the process and IT perspective.

It is evident that the strongest driving force for implementing KBE applications is process improvement through automation. This is explicitly confirmed for the case of small and medium-size enterprises (SMEs) in a study by Cederfeldt and Elgh (2005) who investigated the state of design automation in SMEs. Process-related aspects such as shorter lead times and reduction of labor-intensive tasks scored much higher as driving forces than the establishment of a knowledge base (Cederfeldt and Elgh, 2005). Even though KBE entails knowledge, the knowledge is treated merely as an enabler for the automation. The notion of KBE as a means for managing knowledge is addressed in some cases (e.g. applications 6, 11, 15, 16 and 18) in rather general terms such as a way to address loss of knowledge (Pinfold and Chapman, 1999) or as a way to reduce knowledge bottlenecks to enable access to relevant knowledge when and where it is needed (Sandberg et al., 2008). These are, however, presented mostly as positive side effects and KBE is seldom seen as part of a knowledge management strategy alongside a process improvement strategy.

2.1.3 KBE development tools

Using the definition of KBE development tools in Section 2.1, several IT tools can be considered to implement KBE – far more than what is commercially defined as KBE, i.e. the CAD system-integrated modules. Nevertheless, it can be useful to start off with those tools and investigate their capabilities in order to see whether these functionalities can be found elsewhere.

The CAD system-integrated modules for capturing and reusing knowledge mainly contain functionalities to connect different geometries through mathematical relations. Other useful functions are the possibility to introduce rules. The most common setup for the rules is to evaluate some characteristic of the design and then, based on the outcome, to perform actions. These functions are very useful when geometry-driven elements are evaluated, such as weight, volume, thickness and so on. Which characteristics can be evaluated is dependent on the CAD system, but a rich variety of evaluated characteristics as well as CAD-integrated metadata (for example, ID tags of specific geometrical features) and so on can be used. The trend within the development of CAD systems is to include different kinds of analyses – e.g. regarding mechanics, dynamics, clash detection, and ergonomics. Many characteristics that can be evaluated interactively can also be evaluated by a rule, making it possible to automate even some high-level tasks. Finally, there are possibilities to include checks to see if the model is compliant with standards for holes, fillets, drafting angles and so on. All of these functions are integrated in CAD and thus only related to geometries of mechanical components.

The history of the CAD system-integrated KBE modules is that they used to be stand-alone KBS tools which belong to a special category of tools whose purpose is to support development of knowledge-based systems. They are commonly called shells because they have the generic structure and functions to satisfy the KBS architecture of a knowledge base that is executed by an inference engine. Usually the inference engine is provided by the tool supplier and a structure is given for the knowledge base. The term shell thus means that the developer of a KBS application only has to populate it with the specific knowledge needed in order to obtain a KBS application. In the case of the CAD system-integrated KBE modules, this specific knowledge comprises all the rules and relations that the user defines which are then executed by an inference engine when the geometry is updated. The process of rule execution is not visible to the user. An extensive list of such shells, which existed at the time of writing their book, is provided by Guida and Tasso (1995).

Other IT tools that can embed engineering knowledge are those in the CAE domain, such as tools used for finite element analysis, e.g. MSC Nastran and ADAMS (MSC Software, 2008). Typical knowledge that is embedded in these applications is reuse of load cases, how a component is meshed and so on, which is another kind of engineering knowledge but still knowledge that can support the product development process.

Tools such as Matlab (Mathworks, 2008) have higher abilities to perform tasks similar to those in both CAE software and CAD software, but they do require more programming skills and effort. Even more general-purpose tools, such as Microsoft Excel (Microsoft, 2008), can be used to store and execute formulas and rules similar to those in the CAD system-integrated modules, but lack the ability to analyze engineering-related characteristics. The primary strength of such tools is their availability and ease of use. Finally there is the possibility of using programming languages to create one's own applications in Java or C++. The requirements for programming skills and the effort are higher, but so is the flexibility.

The common characteristic of the above-listed commercial systems (and similar ones) is the possibility to automate tasks through scripts, macros and such. This characteristic is relevant in relation to the definition of KBE given in Section 2.1. If an administrative or repetitive task is performed manually inside a certain tool (or set of tools) no matter which category that tool belongs to (CAD, CAE, PDM etc), it is logical to implement the automation in that tool as well. The prerequisites are that the automation can be implemented (through scripts or similar) and that the necessary knowledge (in the form of rules, relations etc.) can be represented. It is only if these prerequisites are not fulfilled that other solutions should be considered.

2.1.4 Conclusions about KBE applications and development methodologies

The methodological frameworks primarily support the building of an IT application which applies a set of formalized knowledge to automate a specific task. The issue of isolated islands is addressed to a certain extent in the methodological frameworks of KLIC and CommonKADS through the initial steps, in which the process and IT environment is analyzed to ascertain the need for the KBS application and its interfaces. Looking instead at the way potential for KBE is identified, only the KLIC methodology provides support at a general level and from a process and IT perspective. Turning to the knowledge management perspective of KBE, none of the frameworks provide any support other than for the formalization of the knowledge. The recognition of KBE as part of a knowledge management strategy is almost non-existent, apart from the fact that knowledge is seen as a critical enabler and that there are some high-level benefits of formalizing the knowledge in order to reuse it.

Turning to the published KBE applications, it is interesting to see that very few even reflect a methodological approach and that none of them have used (or at least reported the usage of) any of the existing methodological frameworks. Regarding the issue of isolated islands, few applications consider the process and IT aspects which strongly affect the integration of KBE in the product development process. In addition, none of them describe how the need and potential for the published application was identified or derived. Similar observations were made by Bachrach (1997) in a study of how design automation is applied in six different companies, who recognized two approaches in the way opportunities for applications were identified in the observed companies: technology-driven and problem-driven. The technology-driven approach is characterized by trial and error in identifying potential KBE candidates on the part of those knowledgeable in design automation technology but lacking business process knowledge. The problem-driven approach is more rare since people knowledgeable in both business processes and design automation are rare.

A conclusion is that there is a lack of "pull" strategies in KBE development and implementation. Methodological frameworks reflect a "technology push" approach reflected mainly in the way potential for a knowledge-based application is identified. Two of the methodologies, MOKA and Cederfeldt's methodology, explicitly state that this is out of scope. MOKA refers to KLIC and CommonKADS in this respect. These two, in turn, reflect a "solution looking for a problem" approach in the initial phases. KLIC provides an extensive list of characteristics of tasks for which KBS might be applicable along with general indicators of potential. CommonKADS has the ambition to derive a need using the organization, task and agent models but essentially defers back to an approach that consists of listing "typical tasks" suitable for KBSs. Two aspects should be added here. Firstly, even where there is ambition to identify potential it mainly deals with process and IT considerations trying to identify "the task" to automate. Secondly, both of these frameworks are in the more general field of knowledge-based systems, lacking the specifics of product development and engineering.

The lack of pull strategies in the literature describing implemented KBE applications could simply be, as Bachrach (1997) also notes, due to the fact that there are few people, if any at all, in a specific organization who are knowledgeable enough in both business processes and KBE technology to elaborate a "pull" for KBE applications from the business process perspective.

2.2 Knowledge management in product development

This section discusses how knowledge and knowledge management are addressed and what their roles are considered to be in product development. The purpose of this section is to identify key concepts from the field of knowledge management and analyze how they relate to product development, as well as to point out research gaps related to knowledge management in product development.

As briefly mentioned in the introduction, product development is recognized as a knowledge-intensive process where existing knowledge is applied and new knowledge is created (Ullman, 2003; McMahon et al., 2004; Kennedy et al., 2008). In contrast to production, the product development process does not deliver a tangible thing but rather information and data which describe both the product and the process for producing the product. Knowledge is considered to be an important enabler for an effective and efficient product development process (op. cit.) and it is therefore equally important to manage this enabler in a carefully planned way (Clarkson and Eckert, 2005). The case of Toyota's product development system (which has been the main empirical base for the whole Lean Product Development movement) has been a leading example which demonstrates the effects of a product development system that is based on the premise of knowledge as one of the key deliverables of product development (alongside the product). It is argued that this is one of the main factors behind its success (Ward, 2007; Kennedy et al., 2008).

The philosophical and epistemological community has not yet reached a consensus on a definition of what "knowledge" is nor how it should be "managed" (Styhre, 2003). Styhre's (2003) main critique towards the notion of knowledge management is mainly related to the facts that neither of the terms "knowledge" and "management" is defined clearly enough to make "knowledge management" clear. In addition, Styhre (2003) argues that what he refers to as "the mainstream knowledge management community" (such as (Nonaka, 1994) and (Dixon, 2000)) is more concerned with "management", in terms of achieving results and control, than with "knowledge". The internal conflict in the notion of "knowledge management" that Styhre (2003) points out is that management is equivalent to control while knowledge is something flexible that cannot be controlled.

From the perspective of this thesis and from the context of product development, the reason for engaging in "knowledge management" using models and theories from the "mainstream knowledge management community", along with a more pragmatic view on knowledge management in product development, can still be beneficial and have positive effects on the products and the processes in terms of quality, costs and lead times.

2.2.1 Knowledge in product development

It is important to bear in mind that the purpose of this thesis is not to define what knowledge is or to contribute to the fields of epistemology or knowledge management. This subsection aims at describing some models for how knowledge can be seen and categorized from the perspective of product development. The purpose of these descriptions is to discuss some existing models which can facilitate the identification of knowledge in a given product development process, with the aim of recognizing knowledge that is important to manage from the perspective of that particular process.

One way to identify knowledge is by reflecting upon its role in the product development process. Ullman (2003) presents what he refers to as the "value of information" model, as illustrated in Figure 2-7, which is a categorization of different kinds of information and their value for the product development process. The underlying perspective of the product development process adopted by Ullman is that it is based on a series of decisions which make the process progress

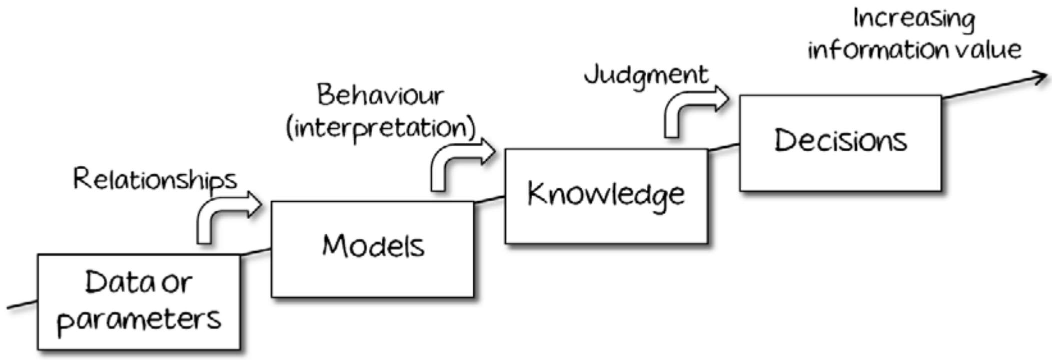


Figure 2-7: Value of information in product development according to Ullman (2003)

forward. The way to recognize knowledge is to identify the information used to enable the decisions and remove the parts that can be considered as models and data in order to isolate what can be considered as "knowledge" important for that process. In a sense, this model of knowledge in product development has large similarities with the model proposed by Ackoff (1989), commonly referred to as the "knowledge pyramid".

Another way to support the recognition of knowledge in product development is through categorizations. Categorizations can be helpful by providing a mental structure as a "thinking support" but should be used carefully in order to not impose too much structure. Two different categorizations are of relevance for this work. One way to categorize knowledge is to divide it into product and process knowledge (Sunnersjö, 1994; Wallace et al., 2005). Product knowledge relates to knowledge about e.g. relations between product parameters, while process knowledge is about activities – e.g. how a simulation is interpreted.

Another way of categorizing knowledge in product development is to divide it into explicit and tacit knowledge (Wallace et al., 2005). The notion of tacit knowledge was originally described by Polanyi (1967) who proposed the existence of knowledge that is deeply embedded inside the human brain and whose effects can be observed, but which cannot be formulated by its owner, at least not without support. As opposed to tacit knowledge, explicit knowledge is knowledge that can be expressed, recorded and stored (Nonaka, 1994).

The notions and definitions of tacit and explicit knowledge are still under discussion in the knowledge management community (Styhre, 2003) and will not be elaborated upon here. It is sufficient to say that a portion of the knowledge involved in a product development process is explicit and can be codified using different kinds of IT solutions. An example of such knowledge are empirically derived design rules which are formulated based on experience of a particular design solution, material, market/customer segment and so on. Another part of product development knowledge is tacit and will manifest itself through an engineer's capability to either skillfully perform design activities or create good design solutions guided by what have been referred to as decisions based on "gut feel" (Raudberget, 2010). Whether the amount of knowledge can be measured or not, or whether one labels explicit knowledge as "knowledge" or "information", is of secondary interest. The primary interests from the perspective of the product development process are:

- The effects it (the knowledge) has on the product development process by enabling better and faster decisions, which in turn appeal to the overall goals of product development to strive for shorter lead times, higher quality and lower costs.
- Where it (the knowledge) resides and how it is managed with the aim of being applied in the most effective way to produce the desired effects.

In line with these, a third way of categorizing knowledge that is of a more pragmatic nature is included in this thesis and illustrated in Figure 2-8. It consists of four categories: know-what, know-how, know-why and know-who (Lundvall and Johnson, 1994). In its purest form it is quite general and provides a framework that is useful as a starting point when assessing knowledge in any process. An example of how this categorization can be applied to the context of product development is demonstrated by Fu et. al. (2006), who have used it to derive a framework for knowledge management in collaborative decision-making in product design.

In this thesis the two categorizations of product/process knowledge and know-what-how-why-who are combined into the model illustrated in Figure 2-8. This combined model constitutes one of the cornerstones in the proposed framework for formulating explicit knowledge management strategies in product development described in Chapter 5. The level of formalization, indicated as a third dimension in Figure 2-8, denotes how formalized each type of knowledge is, and has relevance for the selection of appropriate knowledge management methods and tools which depend on the level of formalization. It should be pointed out that several other categorizations of knowledge can be found in the fields of taxonomies and ontologies in product development (McMahon et al., 2004; Ahmed et al., 2005). The aim of these, however, is to describe the knowledge in a specific field in order to guide the development of IT support for managing the knowledge in that field.

2.2.2 Knowledge management approaches

This subsection describes different strategies for managing knowledge in product development. The primary purpose of these strategies is to guide the selection of appropriate management approaches, methods and IT solutions to best meet the needs to manage the knowledge found in a given organization or process that deals with product development.

Personalization and Codification

These two strategies deal with how the knowledge is managed that is assumed to exist already in the organization in some sense. Codification aims at making explicit as much of this knowledge as possible with the ultimate aim of documenting (codifying) it. The aim is to be able to reuse this knowledge both in time and between individuals. Dedicated methods that embody this strategy are e.g. lessons learned (NASA, 2010), engineering checklists (Sobek et al., 1999),

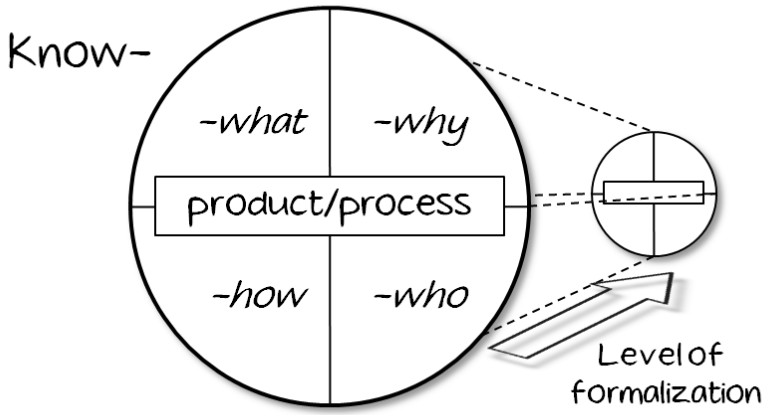


Figure 2-8: The categorizations of knowledge used in this thesis

expert interviews (Hoegl and Schulze, 2005), standards (ISO, 2010), guidelines (Pahl and Beitz, 1996), design/process rules and so on. It is also possible to support codification by including knowledge documentation as part of product/process documentation (Wallace et al., 2005) e.g. in process maps, technical reports, engineering change notifications etc.

The other strategy, personalization, is based on knowledge sharing and reuse by personal interaction and application of knowledgeable individuals. In this strategy the ability to identify "who has which knowledge" is critical and therefore this knowledge is the only one that is codified. Dedicated methods which support personalization are e.g. yellow pages (Hansen et al., 1999), mentoring and apprenticeship programs (Sandberg and Targama, 2007) and special interest groups (Gammelgaard and Ritter, 2005). Personalization can also be enhanced by designing the layout of the work area to enable quick communication and create "natural" meeting points.

McMahon et al. (2004) analyzed the two strategies with the conclusion that both strategies support needs found in product development and complement each other. McMahon et al. (2004) conclude that specific characteristics of different design teams' attributes in terms of size, location and type of product render different needs for knowledge management. These characteristics might call for different combinations of personalization and codification in a specific knowledge management strategy.

These two strategies correlate to a certain extent with the level of formalization mentioned in the previous subsection. The correlation is that a certain level of formalization can only be achieved by codifying the knowledge. Different levels of formalization impose different requirements on the codification. Knowledge that is encoded in rules which are computer-executable exemplify a high level of formalization, while a document template for writing lessons learned exemplifies a lower level of formalization although still implying the use of a codification strategy. Personalization does not require any level of formalization for knowledge regarding know-what, -how and –why, while it may imply a high level of formalization for codifying know-who through e.g. standardized competence descriptions.

Exploration and Exploitation

The terms exploration and exploitation in regard to knowledge management were originally described by March (1991) as a trade-off between allocating resources to activities for seeking new knowledge (exploration) and reusing existing knowledge (exploitation). Different attributes and consequences are credited to both strategies. Exploration is characterized by less certainty and longer time frames in terms of returns on investment, but is necessary for the long-term survival of the company. Exploitation, on the other hand, means more certain short-term returns with apparent limitations of coping with changes which make the reused knowledge obsolete or inadequate.

Revilla et al. (2010) have shown that the adoption of exploration/exploitation in product development practice is related to what they define as the "dynamism" and "complexity" of the business environment. Dynamism refers to the rate of change in the business environment, and complexity refers to the diversity of factors relevant for product development success. The notion of dynamism and how it relates to IT support for knowledge management in product development is also elaborated by Pugh (1996). His notions of conceptually dynamic and conceptually static product designs reflect high and low dynamism, respectively. Pugh (1996) concludes that the use of advanced computer support (such as KBE) is risky for conceptually dynamic product designs because a change in the fundamental product concept leads the digital models to become of limited value. Transferred to the context of this thesis and combined with the findings from Revilla et. al. (2010) and Sunnersjö (1994) (see Figure 1-2), it is implied that the use of KBE (which reflects a codification and exploitation strategy) should be applied in business environment with low dynamism and conceptually static product designs.

2.2.3 Knowledge transfer, roles and motivation

It is important for the implementation and acceptance of any explicit knowledge management strategy that the processes through which the knowledge is captured and applied connect to the business operations in a clear and logical way (Dixon, 2000) while simultaneously supporting the aim of the knowledge management strategy. As Fuxin (2005) notes, the idea of knowledge management has the inherent characteristic that it implies a need for increased documentation in the knowledge-producing end in order to provide benefits in the knowledge-consuming end. Therefore, already from the start, there is a motivational barrier, especially if the benefits are unclear e.g. due to sub-optimization between the organizational entities which make the investment and those which obtain the benefits, or if there is a large time span between knowledge capture and application making it hard to observe an effect. To offset this risk it is necessary to complement the knowledge management strategy with processes for knowledge transfer supported with clearly defined roles, all of which have to be carefully designed to avoid different kinds of motivational barriers that can jeopardize the complete strategy.

The capture and application of knowledge always entails knowledge transfer, which can take place in respect to two different dimensions: across contexts and across individuals. Dixon

(2000) has defined five different knowledge transfer modes. Four of these have been placed in the graph in Figure 2-9 illustrating the two different dimensions. A fifth mode labelled “expert transfer”, where knowledge held by an expert is transferred through consultation, is a kind of “near transfer” and is therefore not included in the graph.

Closely related to knowledge transfer are the roles necessary for it to happen. Markus (2001) identifies three necessary roles in any knowledge transfer or application: the knowledge producer, intermediary and consumer. The producer is naturally responsible for generating the knowledge in the first place and then making it explicit in some form. The knowledge intermediary’s role is to elicit, index, summarize, sanitize and package the knowledge in order to qualify it and ascertain that it is ready for the intended context of application. Finally the consumer is the one applying the knowledge in the way it is intended.

Most of the issues regarding motivation are concentrated at the knowledge-producing end. This comes as a natural consequence of the fact that time needs to be invested and it may or may not benefit those individuals, as noted by Fuxin (2005). Dixon (2000) identifies motivational barriers related to the lacking of four different aspects: time, knowledge, means and will. The lack of time is according to Dixon (2000) a managerial question, but it can also be that a wrong kind of transfer is supported – e.g. the method and roles chosen to support the transfer assume “near transfer” while it is in fact a question of “strategic transfer”. Lack of knowledge implies that the knowledge producers are not able to formulate their knowledge, which could be an effect of poor or inappropriate methodological support. Lack of means implies that the equipment or technological solutions used to support the knowledge transfer are insufficient. Finally, the lack of will is attributed to one or several of the previous lacks according to Dixon (2000), while Markus (2001) states that the lack of will can be addressed through appropriate incentives – either extrinsic (e.g. money, partnership) or intrinsic (e.g. reputation).

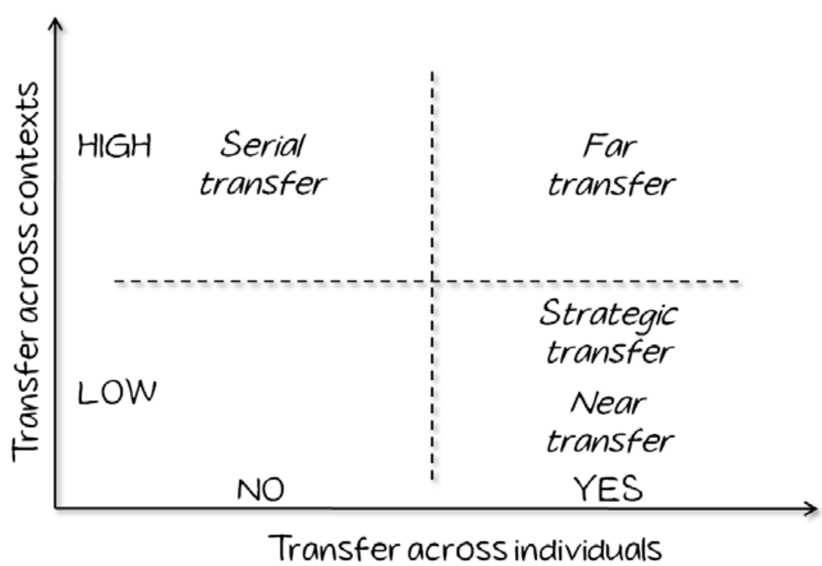


Figure 2-9: Modes of knowledge transfer in the two different dimensions of transfer across individuals/contexts

Motivational barriers exist in the knowledge-consuming end as well. Dixon (2000) has found that, in near transfer, the “not invented here” syndrome can be present and negatively affect any systematic attempt to reuse “someone else’s knowledge”. A similar effect is described by Cummings and Bing-Sheng (2003) as the need to “unlearn” previous knowledge in order to accept new knowledge. Before implementing a systematic approach for knowledge management, Dixon (2000) proposes to investigate the informal networks in which people already talk to “someone else”, find out what knowledge they share and learn why certain pieces of knowledge are needed. By aligning the explicit knowledge management strategy to an already present social pattern, it is far more likely to be accepted than one which totally ignores the pattern. Another issue affecting motivation in the consumer end is illustrated in the quote “we have already tried ‘knowledge management solution ‘X’ and it was not used” (Dixon, 2000) which is largely attributed to a misalignment in previous attempts of knowledge management. Finally, the lack of means or technology for accessing knowledge can also cause low motivation in the consumer end (Dixon, 2000; Markus, 2001).

In the context of the product development process and its improvement, the three transfer modes of “serial transfer”, “near transfer” and “expert transfer” are of interest. The modes of “far transfer” and “strategic transfer” are relevant for e.g. technology transfer from research into product development. Even more important, however, is how the necessary roles are divided among the individuals involved in an explicit knowledge management strategy, an aspect which is not at all present in the literature from the area of product development. The need for assigned roles in knowledge management is only partially addressed by the methodologies for developing KBE applications, which heavily rely on the role of a “knowledge engineer” who plays the role of the intermediary but whose actual purpose is to develop and maintain the KBE application from an IT perspective.

Looking at the way motivation for knowledge management is addressed in literature from the field of design research, some interesting observations can be made. In general it can be said that the widely adopted perspective is to strongly couple motivation to both capturing and applying knowledge in direct relation to the ease of doing so. In the case of knowledge capture the term “ease” can be said mostly to relate to “lack of time” and “lack of means”, and is attributed to factors such as:

- IT system and its user interface (Chakrabarti et al., 2005; Greg et al., 2009; Pavković et al., 2009), implying that an easy to use IT system positively impacts user motivation.
- Appropriate knowledge taxonomy (Ahmed, 2005) implies that a proper taxonomy eases the identification of relevant knowledge in the application phase and has positive effects in the capture phase through better support for contextualization.
- Knowledge about methods and IT solutions among users (Salehi and McMahon, 2009) implies that increased knowledge about knowledge management also increases the motivation to engage in it and has elements of “lack of knowledge” from Dixon (2000).

- Purposefulness and effectiveness in the product development process (Boart et al., 2006) refers to the direct benefit in the design process which motivates in the application phase.
- Integration of IT applications and product development processes (Ćatić, 2009) implies that higher integration reduces the risk of having outdated knowledge in the IT applications, avoiding the risk of decreasing the motivation in the application phase.
- Alignment of IT support for knowledge capture with the designer activities (Brissaud et al., 2003; Bracewell et al., 2004) implies higher motivation due to less interference with the design process, primarily to address "lack of time".

The issue of knowledge management is approached with the perspective that motivation of individuals is realized by making knowledge management as invisible and non-intrusive as possible regarding its integration in both the IT support and the processes. Motivation as an explicit aspect is seldom mentioned, especially at the level of individual designers. One exception exists (Weilguny and Gerhard, 2009) where motivation is addressed through a detailed analysis of the organizational change management aspects in the implementation of a KBE application (number 18 in Appendix A) in the application phase. Some contributions explicitly address motivation at a company level by establishing a "business case for knowledge management" relating it to shorter lead times, increased product quality, lower design cost and other strategic goals of product developing companies (Cederfeldt and Elgh, 2005; Gardoni and Dudezert, 2005).

2.2.4 Conclusions regarding knowledge management in product development

The notions of knowledge and knowledge management in product development have been discussed by several authors in the field of product development. McMahon et al. (2004) and Revilla et al. (2010) provide contributions regarding how different aspects of product development (e.g. team distribution, type of product, type of market etc) favor different knowledge management strategies. Fu et al. (2006) contribute with how the general framework of know-what-how-why-who is applied to collaborative decision-making in product development. Though valuable, these contributions merely provide insight into how bits and pieces from the field of knowledge management can be applied to product development. There is still a research gap regarding how to proceed when formulating an explicit knowledge management strategy in the context of product development. One attribute of the contributions by McMahon et al. (2004) and Revilla et al. (2010) that is important to highlight is the fact that they essentially reflect a "pull approach" in which the knowledge management strategy is obtained based on the characteristics of the product development organization or process.

Comparing further the knowledge management literature with how knowledge management is approached in product development literature, it is clear that there is a gap in how aspects such as roles, strategies, methods and processes are addressed. The literature presented in this section primarily highlights the need for an explicit knowledge management strategy to be guided by a structured approach embodied in a process for knowledge capture and application supported by defined roles and appropriate methods. The fundamental message is that a knowledge management initiative is not going to naturally integrate and embody itself in the organization simply through an expressed managerial ambition. The fact that knowledge management can be generally motivated at a company level as contributing to shorter lead times, increased product quality and lower costs does not imply that these aspects also motivate each individual (which is another aspect that is lacking in product development research literature).

2.3 Product development processes

In the following subsections some basic variants of the product development process are described from a knowledge perspective. Processes for new product development which are commonly used as main reference points for both academic (Andreasen and Hein, 1987; Pahl and Beitz, 1996; Ullman, 2003; Ulrich and Eppinger, 2008) and industrial purposes (VDI 2221, 1993) are analyzed from a knowledge point of view. In addition, the process for product development defined in the Lean Product Development framework (Morgan and Liker, 2006; Ward, 2007; Kennedy et al., 2008) is described due to its explicit focus on knowledge as a deliverable. The processes of configuration design and engineering change management, which in comparison can be considered as less open-ended, are highlighted here because of their focus and dependence on explicit knowledge.

A common aspect of these processes is that they apply a stage-gate model (Cooper, 2008), and typically the output is considered to be the product and the information describing the product to the extent that it can be manufactured, legislated, marketed, sold, used and serviced. It can be argued that the information output from the product development process contains product know-what in order to communicate the aspects of the product design necessary for the above mentioned. To some extent also product know-how is contained within the output information where necessary, e.g. how the product is assembled in manufacturing, disassembled for maintenance, configured for sales etc. In the context of this thesis, however, product development processes are observed as being producers and consumers of all of the eight types of product and process knowledge; see Figure 2-8. This perspective is necessary in order to establish the view of product development processes as the specifying and receiving end (or "customer" in more pragmatic terms) of an explicitly formulated knowledge management strategy.

2.3.1 New Product Development

The traditional view of the new product development process is that it starts with an idea or a market need for a product, and then through a sequence of activities of synthesis and analysis a final design emerges which is validated and verified to fulfill all the requirements on it; see Figure 2-10 which is used as an illustrative example from Andreasen and Hein (1987).

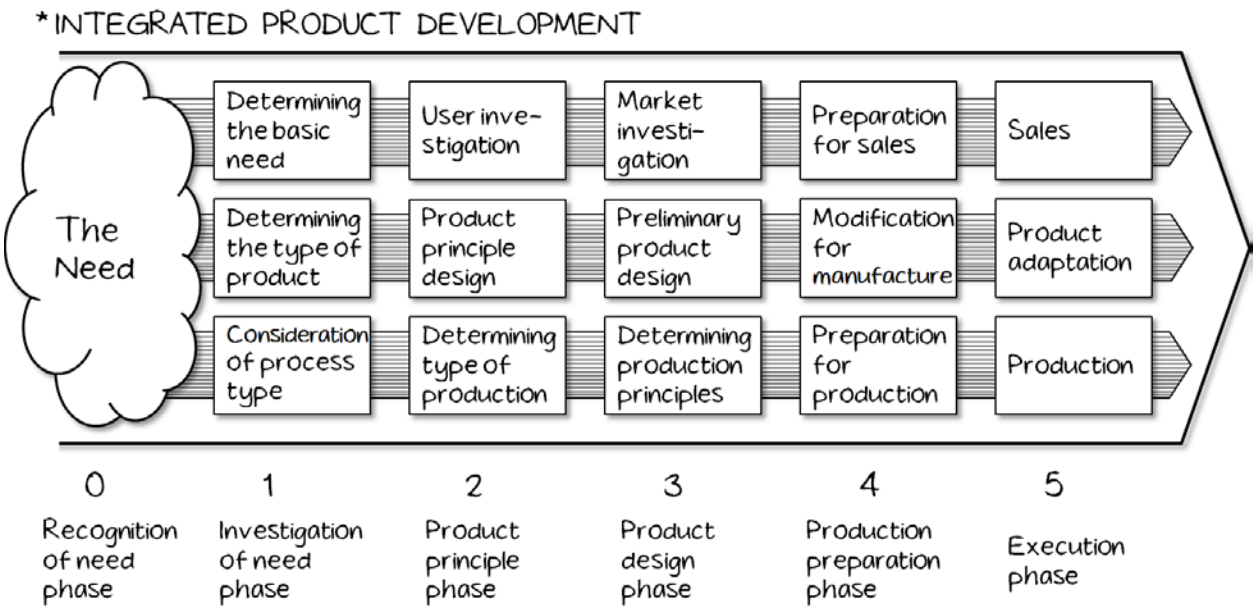


Figure 2-10: Example of a product development process framework (Andreasen and Hein, 1987)

The core of any activity in the sequence is to increase the concretization of the requirements, the design or the performance of the design. In the end the design is validated and verified. By this stage, the design is described in enough detail that it can be manufactured and later used and serviced. The details of how to go about planning, sequencing and executing this process have been addressed by many authors in similar process frameworks (Andreasen and Hein, 1987; Pugh, 1990; Suh, 1990; Cross, 1994; Roozenburg and Eekels, 1995; Pahl and Beitz, 1996; Ullman, 2003; Ulrich and Eppinger, 2008). Based on experiences of teaching, being, managing, studying and consulting designers these authors give their perspectives on what to do, how to do it, when to do it and how to support it in terms of methods and IT solutions, the details of which lie out of scope for this thesis.

It is clear that knowledge is not defined as an explicit deliverable that needs to be managed in an equally explicit fashion in any of the processes. The focal points are the design solution, how the solution is analyzed and how it is documented enough to be produced, marketed, sold, used and maintained but not necessarily reused. The documentation can contain elements of product know-what and know-how but only to the degree that the documentation requirements are satisfied. Issues like product and process improvement using knowledge created and captured in previous projects or iterations are not addressed. Some of the exceptions are Ulrich and Eppinger (2008) who recognize a need to perform a post-project assessment and evaluation as a kind of lessons learned. The issue, however, is addressed only briefly with relatively little guidance.

As mentioned earlier, Ullman (2003) provides some reasoning about the role of knowledge in product development and how it relates to product development, but does not really address a need to make knowledge an explicit deliverable and manage it as such. In product development frameworks, such as integrated product development (see Figure 2-10) (Andreasen and Hein,

1987) or concurrent engineering (Prasad, 1997), it could be argued (by e.g. (McMahon et al., 2004)) that knowledge is implicitly reused by making sure that individuals with knowledge regarding specific product lifecycle aspects are involved in the design, thus reflecting a personalization strategy. Towards the other end, Design-For-X methods could be said to represent a codification strategy in support of concurrent engineering (Huang, 1996). Even though purposeful, this is still far from providing support for the establishment of an explicit knowledge management strategy based on the needs of a specific organization where knowledge is recognized as a deliverable that needs dedicated activities, roles and methods for its management.

2.3.2 Lean Product Development

The term "lean" in Lean Product Development comes from Lean Manufacturing (Liker, 2004) which has revolutionized production management. The cornerstone of "lean" is to focus on different kinds of waste in any process and systematically address reduction of it through application of methods and tools, management practices and organizational mindsets. Liker (2004) describes seven types of waste related to over-production, unnecessary movement of material, waiting, error correction and so on.

Focusing on the specifics of product development, Morgan and Liker (2006) have chosen to describe and exemplify similar kinds of waste. These can be defined largely to be operational kinds of waste which arise in the execution of product development processes such as not having access to the right information, performing unsynchronized concurrent tasks and waiting for decisions. It can be argued that these kinds of waste can be addressed with methods and tools for information management, project planning and execution, process modelling, virtual teams, task automation, rapid prototyping and virtual product development found in various branches of design science.

Ward (2007) and Kennedy et al. (2008), however, have chosen to define another type of waste in product development which causes all other wastes. It is waste of product and process knowledge; see Figure 2-11. The authors state that since product development is a knowledge-intensive task, it produces knowledge as much as it produces designs and information about the

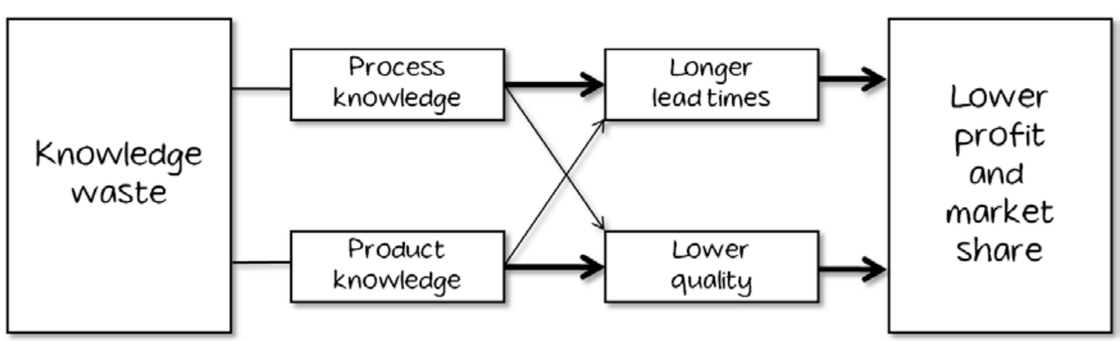


Figure 2-11: Waste of product and process knowledge and its main consequences adapted from Ward (2007)

designs. While the methods and tools proposed in the traditional product development paradigm address the waste of information and time, they do little or nothing to address the waste of knowledge which, according to these authors, is the big waste in product development. The case in point is Toyota, whose allegedly consistent product development successes in terms of quality, cost, lead times and profit are attributed to a systematic capture and reuse of product and process knowledge.

Seen from a knowledge perspective, the main distinction of the Lean Product Development framework is the explicit recognition of knowledge as a deliverable and a consumable in the product development process. This facilitates the establishment of a documented "knowledge baseline" between product development projects; see Figure 2-12. The baseline makes it easier to mitigate risks early in product development projects, allocate resources in relevant areas depending on knowledge gaps, and enable continuous improvement of both the product and the process. In short, the following elements in the framework constitute the foundation of its knowledge orientation and enable the establishment of a knowledge baseline. Some of the elements are methods, some are roles and some are principles.

Chief engineer

The role of the chief engineer is to see that a thorough understanding of the customer needs and wishes is ensured as well as an anticipation of unspoken needs the customer might have.

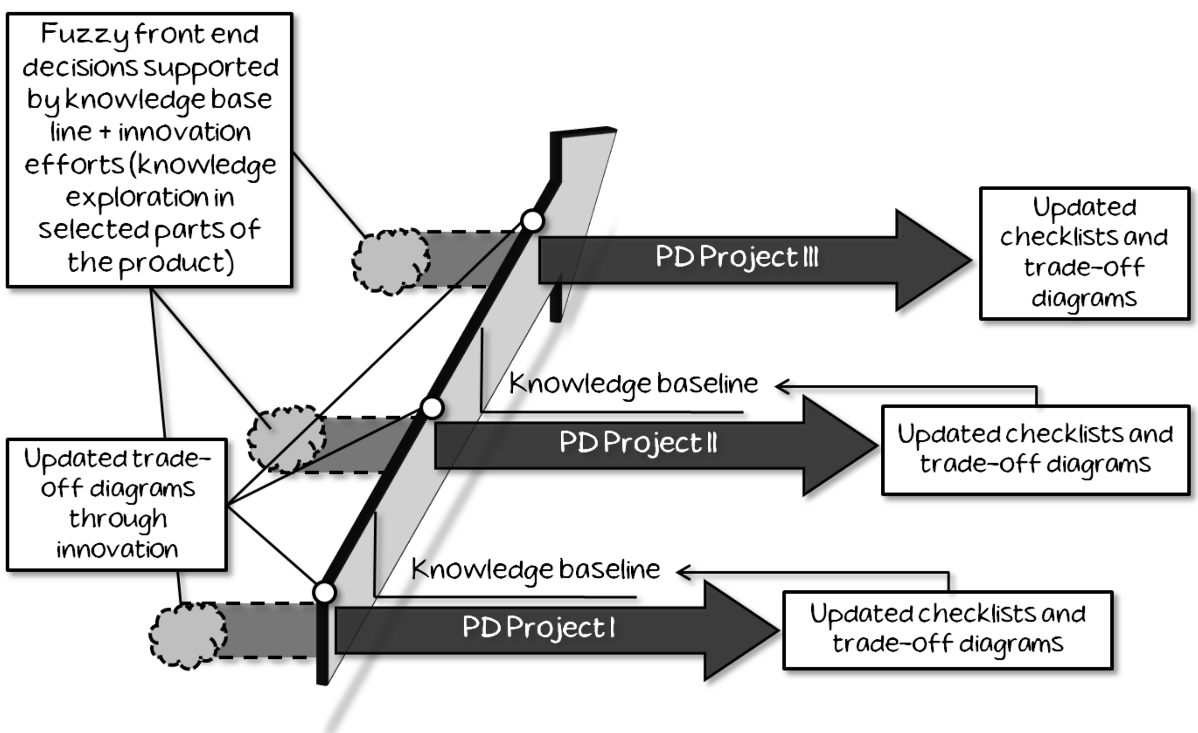


Figure 2-12: A schematic illustration of how knowledge is developed in and between product development projects in the lean product development paradigm (Kennedy et al., 2008)

The chief engineer does this by “becoming the customer”, and stories about chief engineers who live up to 6 months together with the intended customers are described. From a knowledge perspective the chief engineer acquires a lot of explicit knowledge about how the product is used and how it could be used. More importantly, however, the chief engineer acquires a lot of tacit knowledge by using the product, observing the customer and “being the customer”. The application of this knowledge is done later in the decisions regarding the design of the car, where the chief engineer is involved in most of the system-related decisions (which have high impact on the customer’s experience of the car).

Going to the source (“Genchi genbutsu” or “Gemba”)

This principle states that the resolution of a problem requires first-hand experience of that problem. If the assembly of a component is troublesome, the designer of that component should not solely rely on a report but try to assemble the product him/herself in an environment as close as possible to the real one (preferably at the assembly line). This principle fosters decisions to be based on first-hand information and experience because any secondary information or data is in some respect “filtered” by someone. The role of the chief engineer is a primary example of this principle, where the person who is supposed to manage the complete product development project does not solely rely on specifications coming from a marketing or product planning department but uses his/her own experience.

Continuous improvement (“Kaizen”) and standardization of practices

These principles state that as soon as one realizes a better way of doing something one has to propose it in order for this new way to be the new standard way. This in turn is based on the premise that once a standard way is decided upon everyone follows it until someone gets a new idea of how to improve it again. These principles are primarily related to process knowledge but can be applied to product knowledge as well. They simply mean that no good idea for improvement goes wasted, and the principles are thoroughly backed up by a process and methods for how each idea is taken care of.

Visualization

Visualization is a general principle and is commonly referred to in the context of visual planning and scheduling. It has however a strong influence in the decision-making process where the knowledge upon which decisions are based has to be as explicit and visual (through graphs, pictures, models, prototypes etc.) as possible. The only exception is the chief engineer who is allowed to make decisions based on “gut feel” from experience of being the customer.

Knowledge hierarchy and apprenticeship

The idea of knowledge hierarchy is a principle which states that the person who becomes a manager should have a high level of knowledge regarding the work and the output of the group he or she is managing. The role of the manager is to be the mentor whose primary responsibility is to elevate and develop the knowledge of the group through a kind of apprenticeship.

Testing

Testing is a method that is used not only to verify a design in relation to requirements, but is primarily applied prior to product design as an engine for product knowledge exploration where insights about design parameters, relations between parameters and system effects are gained. An example of this is "testing to failure" where the design is exposed to the test beyond the point of approval to see when it fails and why.

Trade-off diagrams and A3 reports

These are used to support the problem-solving process, document it and simultaneously capture product knowledge. They constitute both a supporting method and a standard format which applies the visualization principle and aligns knowledge documentation with the design process.

Engineering checklists

These are used to support the detail design process, and simultaneously to capture and reuse process knowledge so as to ensure that issues related to the product lifecycle, interfaces and legislation are considered. This method appeals to the continuous improvement principle by making knowledge capture and application simple and easy to use.

Using these elements the organization can, at any given time, know what is known about principal product solutions and processes in terms of capabilities, ranges of validity and so on.

2.3.3 Configuration Design

Configuration design can be defined as the design activity in which the artifact being designed is assembled from a set of predefined components that can be connected in certain ways (Mittal and Frayman, 1989). Configuration design supports the configuration process which outputs a configured product as part of the sales-to-order process of any company with configurable products. The main output of the configuration design process are the rules and constraints governing the configuration process which in the context of this thesis can be labeled as "configuration knowledge". As opposed to the processes mentioned in Section 2.3.1 regarding new product development, configuration design has knowledge as its only deliverable in the form of rules and constraints. This perspective is reflected by Soininen et. al. (1998) who proposed an ontology for the area of configuration. Three main categories of configuration knowledge are identified by the authors:

1. **Configuration model knowledge**
   Specifies entities and their properties along with rules on how the entities and their properties can be combined. Configuration model knowledge specifies the set of correct configurations of a product with respect to the requirements.

2. **Configuration solution knowledge**
   Specifies a configuration in sufficient detail, i.e. what a specific product instance must be like.
3. **Requirements knowledge**
   Specifies requirements on the configuration to be constructed. This knowledge can be specified with the same concepts as the previous two kinds of knowledge.

A further indication that configuration can be considered a rather knowledge-intensive process is the fact that two of the KBS methodologies described in Section 2.1.1, KLIC and CommonKADS, use configuration as a typical domain of application for knowledge-based systems. CommonKADS even provides a template for the worksheets involved in their KBS method. Stumptner (1997) also provides an overview of knowledge-based system methods and techniques applied in the area of configuration discussing the different ways of addressing issues of segmenting between domain knowledge and control knowledge in the inference engine as well as in representing configuration knowledge. Mesihović (2004) has investigated how PDM systems can support the configuration process, and he recognizes knowledge as an integral part of the process being both an input and an output of configuration design.

Apart from (Soininen et al., 1998) most of the contributions in configuration design deal with managing, executing and validating configuration rules using different types of methods and IT tools. The amount of contributions regarding methods for elicitation of configuration rules is smaller (Soininen et al., 1998; Tidstam and Malmqvist, 2011); in knowledge management terms, these would translate into methods for knowledge capture. One of the common issues in knowledge management, the issue of motivation for knowledge capture and application, is not an issue in the context of configuration design. This could be due to the fact that knowledge is the main input and output of the process; thus there is no point in questioning either its capture or reuse. In addition, it should be noted that the term knowledge in configuration design mainly denotes product knowledge (i.e. rules and constraints). Process knowledge is largely included in the discussion regarding problem-solving techniques applied in the inference engines which execute the rule base.

2.3.4 *Engineering Change Management*

Engineering changes are defined to be all kinds of changes which occur after a design has been released (Huang and Mak, 1998; Jarratt et al., 2011). The changes usually induce a series of other changes both to the design of other components in the product and to downstream processes along the product lifecycle. Engineering change management is the process by which the change impact is managed in terms of analysis of a change’s propagation, its approval and follow-up of the sub-sequent changes which result from the original change. It is recognized as a rather complex process in terms of involved people, activities, their sequence and their information needs (Huang and Mak, 1998; Pikosz and Malmqvist, 1998). In addition, Joshi et al. (2005) and Lee et al. (2006) conclude that the engineering change process is knowledge-intensive and that proper support of it requires elements of knowledge management on top of information and workflow management.

Supporting tools for engineering change management were mainly paper-based with occasional computer support, in which case the support is used for keeping track of the process and versioning of documents. In their study of engineering change practice in the UK industry, Huang and Mak (1998) identified a set of needs related to computer support to provide control of the ECM process and ensure data availability and validity. Most of these were in turn related to the need for a formal management of the information and the workflow associated with an engineering change. Among other needs the authors identified, were needs to make sure that the right people were involved in the change process and that those people have access to all the information they need to carry out their tasks of accessing the details about the change, the change's history, prioritize among current changes, work out solution alternatives and evaluate the effects of a change. Similarly, Pikosz and Malmqvist (1998) proposed the use of workflow and data management in commercial PDM systems as a solution to support the ECM workflow, information and document management.

In more recent work, Joshi et al. (2005) conclude that only managing the workflow and versioning documents, as ECM support in modern PLM systems does, is necessary but not enough since such solutions do not take into account the fact that ECs are different from case to case. They propose a dynamic workflow driver implemented in the PLM systems that also captures and reuses process knowledge about different ECs to continually improve the dynamic workflow driver. The idea to capture and reuse knowledge generated during ECM was also implemented by Lee et al. (2006) in regard to product knowledge. The product knowledge about relations and dependences discovered during each case of engineering change is captured along with its context. Using the context, the knowledge is indexed and structured using a semantic web solution, in order to facilitate a more accurate search in the reuse to support decision-makers in predicting future change impacts.

2.3.5 Conclusions about product development processes

There is a lack of focus on explicit knowledge management among the established theories and frameworks for product development processes. The focus on managing workflows and documents in ECM could reflect a personalization strategy in which knowledge is implicitly managed by involving the "right" people depending on the engineering change request. A similar approach is implicit in the recommendations set forward by e.g. Ulrich and Eppinger (2008) when they speak of "appropriately" compiled design teams with individuals from different engineering functions and product lifecycle phases applying the knowledge from their respective domain to affect the product design. In a similar way the same kind of strategy could be assumed as being adopted in the frameworks of concurrent engineering and integrated product development as discussed in Section 2.3.1. The issue that is highlighted here is that none of the respective contributions refer to this as an intentionally chosen strategy, thus leaving the status of knowledge management rather undefined in those processes. Configuration design is the exception to the rule because the fact that it explicitly manages knowledge could be due to the circumstance that knowledge in the form of configuration rules is its only deliverable.

By contrast, the Lean Product Development framework has an explicit approach to knowledge (Ćatić and Vielhaber, 2011). The framework claims that its focus on product and process knowledge management is the only way to achieve truly effective product development processes. Whether this is true to the extent claimed by certain authors (Kennedy et al., 2008) is left for the future to decide, but the approach does at least have a clearly defined strategy and frame of principles, roles, methods for managing product and process knowledge within projects as well as between projects.

2.4 Business Process Redesign

Observing KBE as a process improvement method prompts one to look into the area of business process redesign in order to position KBE in relation to other methods and tools in this field. Being a business process, product development process improvement is in fact concerned with applying models and methods from business process redesign, which in turn belongs to the even more general field of operations management (Davenport and Short, 1990). Davenport (1990) lists the following objectives as most likely for a business process redesign:

1. Cost reduction
2. Time reduction
3. Output quality
4. Quality of worklife/Empowerment

A note of clarification on the last objective is that it concerns motivational factors in that a process redesign always affects the quality of work life and empowerment of the process participants either positively or negatively, and therefore this can be used as the sole objective of a process redesign. Davenport states that even though cost reduction is important, using it as the sole objective can never be feasible, as it might trigger trade-offs related to time, output quality and quality of work life which cannot be accepted by other stakeholders in the process.

2.4.1 Methods for Process Improvement

As part of e.g. the ISO 9001 framework, there exist methods aimed at general process improvement that can be applied to any process. One rather simple method that supports the initial phases is a gap analysis (Hvam et al., 2008) which starts with a documented current state of the process and a reference to which the current state is compared. The comparison is done in the aspects of existence and utilization of activities and tools to realize certain goals, which in the case of ISO 9001 (Mutafelija and Stromberg, 2003) are mainly aimed at the third objective of process redesign, output quality.

The Capability Maturity Model (CMM) (Persse, 2001) guides the assessment and improvement of processes, and stems from software development. It is divided into five levels of maturity:

1. **Initial**
   Entry level of any process regardless of its state, describing the current state.
2. **Repeatable**
   Implies that the state of the process is not ad-hoc and can be executed again. It implies the use of e.g. requirements management, project planning, tracking and oversight, quality assurance, configuration management and subcontractor management.
3. **Defined**
   A set of best practices exists for all the repeatable aspects. The best practices are established, documented and followed to ensure little or no variance in execution. It implies the use of peer reviews, process training and a process focus in the organization.
4. **Managed**
   Implies that the process is repeatable, best practices are followed and outputs are measured to enable continuous improvement.
5. **Optimizing**
   Implies that the first four levels are maintained and continuously improved.

Thomas and McGarry (1994) distinguish between top-down and bottom-up oriented process improvement. The ISO and CMM frameworks are defined as top-down oriented because they rest on the assumption of the existence of universal practices which are required and beneficial to all software development. Process improvement then implies the reduction of dissimilarities between the current state of the process and the reference state. A bottom-up oriented process improvement instead takes its start in the unique performance, problems and characteristics of a particular organization and develops the process to better support the organization in realizing its specific goals. An approach reflecting a bottom-up orientation is Plan-Do-Check-Act (PDCA) method, also called the Deming cycle, which is commonly cited in literature from Lean Product Development as an enabler for continuous improvement. (Kennedy et al., 2008).

Within the field of design science, a method called Design Structure Matrix (DSM) is used to model dependences between tasks in a process represented in a two-dimensional matrix; see Figure 2-13. The simplest operations that can be performed on a DSM are:

1. Identification of iterative loops.
2. Identification of tasks that are independent and can be made concurrent.
3. Identification of tasks that need to be reordered for better product quality, shorter lead time or better resource utilization.

Another method used for modeling dependences between tasks and information is called Signposting (Clarkson and Hamilton, 2000) and is more detailed than DSM but also more demanding to execute. Compared to DSM, Signposting models the dependences between tasks not only as binary X's but identifies exactly which design parameters constitute the dependences and their confidence levels. Such an approach makes it possible to identify chains of activities which increase the confidence levels of product parameters in the quickest way, or identify

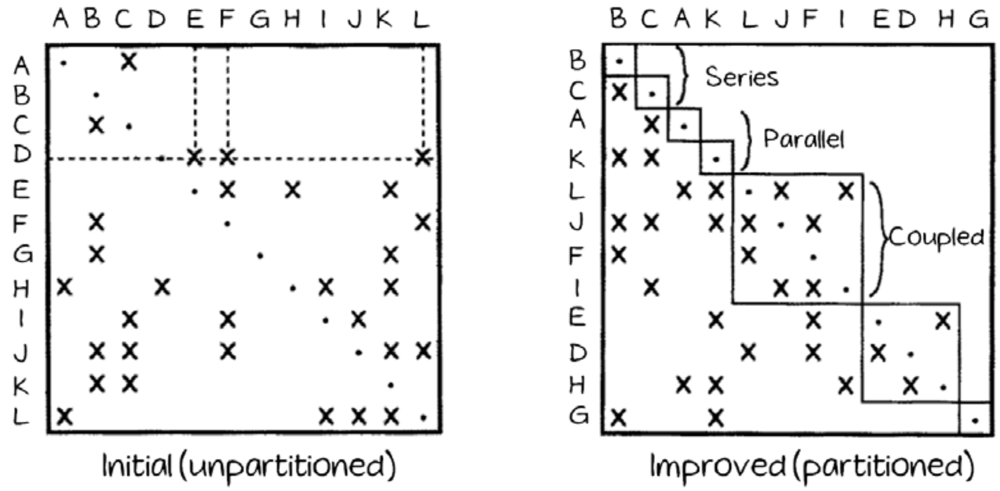


Figure 2-13: Left: the initial process modeled in DSM. Right: the improved process. (Eppinger et al., 1994)

critical chains of activities depending on which product parameters need to be focused upon. An approach with a similar aim, however, has also been achieved by modifying DSM (Isaksson et al., 2000).

2.4.2 IT and Process Improvement

The field of enterprise architecture indicates that the relation between IT and processes is a rather complex one. Regarding the specifics of business process redesign and IT, Davenport (1990) states that there is a recursive relation between the two. The redesign of the business process motivates the application of IT and simultaneously IT capabilities can inspire and affect the redesign of the business process.

IT capabilities that can support business process redesign are defined by Davenport (1990) as:

- **Transactional –** transform unstructured processes into routinized transactions
- **Geographical –** transfer information rapidly making processes independent of geography
- **Automational –** replace or reduce human labor in a process
- **Analytical –** bring complex analytical methods to bear on a process
- **Informational –** bring vast amounts of detailed information into a process
- **Sequential –** enable changes in sequencing to enable concurrence among tasks
- **Knowledge management** – enable capture and reuse of knowledge for process improvement
- **Tracking –** allow detailed tracking of task status, input and output
- **Disintermediation –** connect two parties in a process without an intermediary

The recursive approach advocated by Davenport has the viewpoint of business process improvement as an idealistic joint venture between those responsible for IT and those responsible for business process improvement. The reality, however, tends to favor either an IT-driven approach of "technology push" (Zimmerman et al., 2008) or a process-driven approach of "technology pull" (Svensson, 2003). At its extreme the push approach offers little chance to optimize the process outside of the frame of possibilities provided by a chosen IT solution while the pull approach risks ending up in an IT environment which is hard to manage and integrate due to heterogeneity. In the technology push/pull dimension, Zmud (1984) notes that a pull approach has a higher probability of success. Zmud (1984), however, states that a successful process innovation is most often based on a need and a simultaneous emergence of a means for resolving that need, which also reflects the recursive approach advocated by Davenport (1990).

2.4.3 Conclusions regarding business process redesign

Taking the viewpoint of KBE as a method for process improvement, according to the definition given in Section 2.1, and comparing it to CMM, KBE can be considered as a method for enabling the higher levels of process maturity. It contributes to the systematic capture and reuse of product and process knowledge related to best practices. In relation to the PDCA cycle, KBE can be used as one of the possible solutions considered in the “Do” step of PDCA to solve an identified problem. The conclusion is that KBE can be applied both as a top-down and as a bottom-up oriented method.

Taking the viewpoint of KBE as an information technology for process improvement and comparing it to the different capabilities of IT listed by Davenport (1990), it is clear that the main capabilities of KBE are the abilities to automate and manage knowledge (which are the two aspects of KBE reflected in this thesis). As was mentioned in Section 2.1, automation is the main capability of interest and it appeals to three of the four objectives of business process redesign listed by Davenport (1990): cost reduction, time reduction and increase in output quality.

Combining the dimension of top-down/bottom-up focus in methodological approaches to process improvement with the dimension of IT-supported/IT-driven process improvement results in the graph in Figure 2-14. The top right corner is labeled “technology push” because it is driven by IT capabilities and the improvement of the process is initiated from outside the process. The opposite corner is “technology pull” implying that the need for process improvement is coming

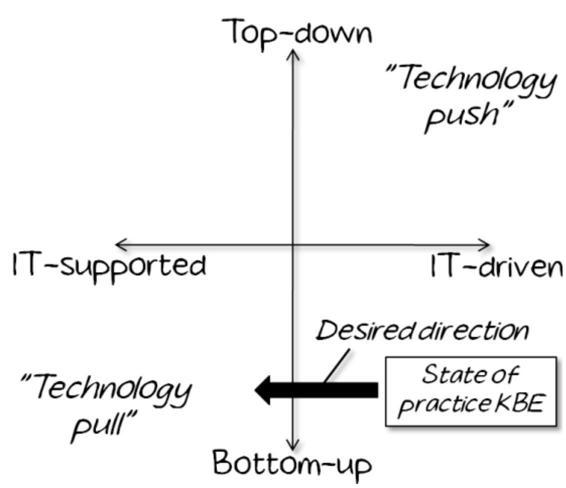


Figure 2-14: The two dimensions of process improvement with current state of KBE mapped out and the desired direction from the perspective of this thesis

from inside the process and is looking for IT solutions based on this need. As implied in Section 2.1.2, the KBE applications summarized in Appendix A all are based on bottom-up process improvements since their implementation addresses needs which originate from within the process, and they bear elements of technology push since they are essentially IT-driven with focus on demonstrating the technologies related to KBE in specific cases.

2.5 Product Lifecycle Management (PLM)

There is no commonly accepted definition of exactly what is embraced by the term product lifecycle management. There exist several definitions, one of which is chosen here due to its generality and ability to embrace PLM both as a technology, as a method and as a process:

> *Product Lifecycle Management (PLM) is a strategic business approach that applies a consistent set of business solutions in support of the collaborative creation, management, dissemination, and use of product definition information across the extended enterprise from concept to end of life – integrating people, processes, information and business systems (CIMData, 2002).*

It can be argued that PLM is an enabler for concurrent engineering and integrated product development, ensuring that information and tools needed to feed the processes with information from different lifecycle phases (manufacturing, aftermarket, marketing, suppliers) or from different technology domains (mechanics, electronics, software) are available. An illustration of PLM is given in Figure 2-15 by Malmqvist (2008), where the business tools (engineering tools) are related to the processes.

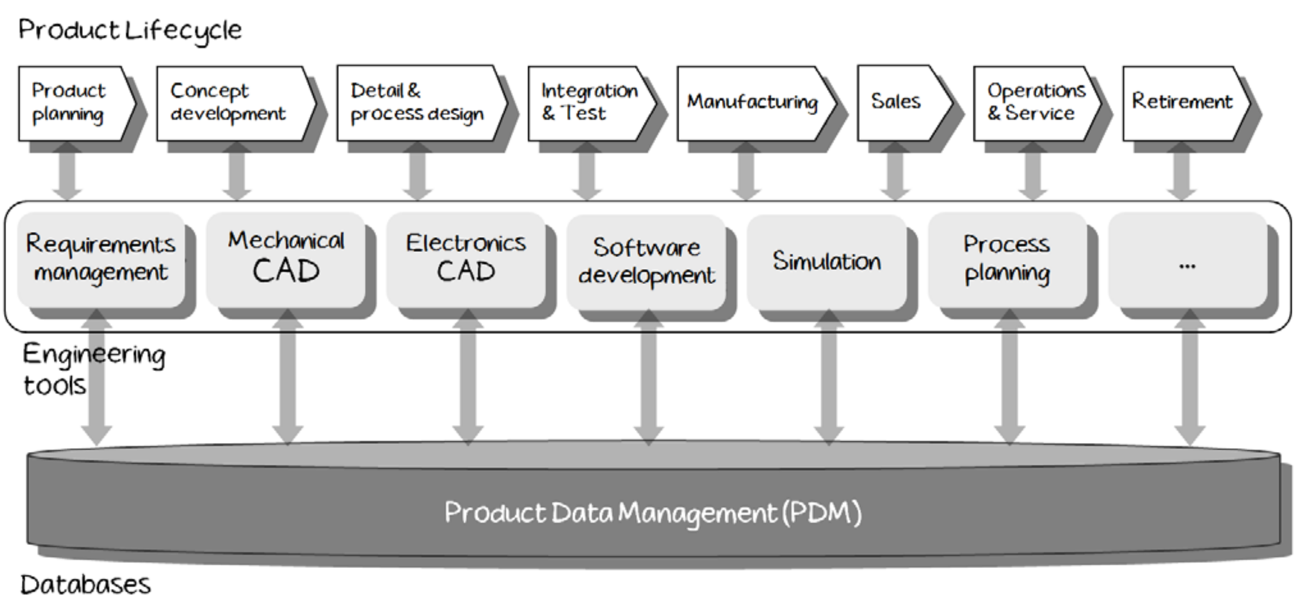


Figure 2-15: Product Lifecycle Management (Malmqvist, 2008)

What is not visible in the figure is the fact that the engineering applications and the processes share the same information model of the product. Otherwise the integration of the tools into the processes would not be realizable. This issue is further detailed in the next section dealing with enterprise architecture and PLM architecture.

2.5.1 Enterprise architecture

The first ideas concerning the concept which today is referred to as the enterprise architecture were published by Zachman (1987) who, based on experience from IBM, defined a framework today referred to as the Zachman framework for enterprise architecture. The framework aims at providing a placeholder for the different views of and relations between the elements of an enterprise's process and IT environment. It makes sure that the same architectural principles govern all the components, no matter whether they are computers, people, information or activities, in order to cooperate in harmony.

After Zachman (1987), several generic enterprise architectures have been proposed such as The Open Group Architecture Framework (TOGAF) (Schekkerman, 2003). Harmon (2004) argues that there are process-centric and IT-centric ways of approaching enterprise architecture. In the process-centric approach, processes and activities have a leading role, with aspects such as process management, optimization and profitability being the main concern, using IT as an enabler. The IT-centric approach aims at modeling and connecting the various IT models and resources that must interplay in the IT environment. Creators and users of IT-centric approaches reside solely in the IT community.

2.5.2 PLM architecture

A PLM architecture can be considered as a kind of enterprise architecture, where the different layers of the process and IT environment deal with product-related processes, information and applications. In an enterprise whose main focus is on developing, manufacturing, selling and maintaining products, this means that a majority of the process and IT environment is embraced by the PLM architecture. In Figure 2-16 a generic PLM architecture is illustrated and populated with some of the more PLM-specific terms in the following layers:

Business strategy and objectives are derived from the overall company visions and strategies. This layer answers the question "What is to be realized by the company?".

Business processes are the activities that need to be performed in order to realize the business objectives and strategies. This answers "What needs to be done to realize the objectives?".

PLM workflows are the detailed activities for how the engineers are going to work in the PLM environment. This answers "How do PLM solutions support the engineers' activities?".

Strategic capabilities are general capabilities in the PLM environment needed in the workflows and essential for the business processes to be performed. This answers "What general functionalities are needed in the PLM environment?".

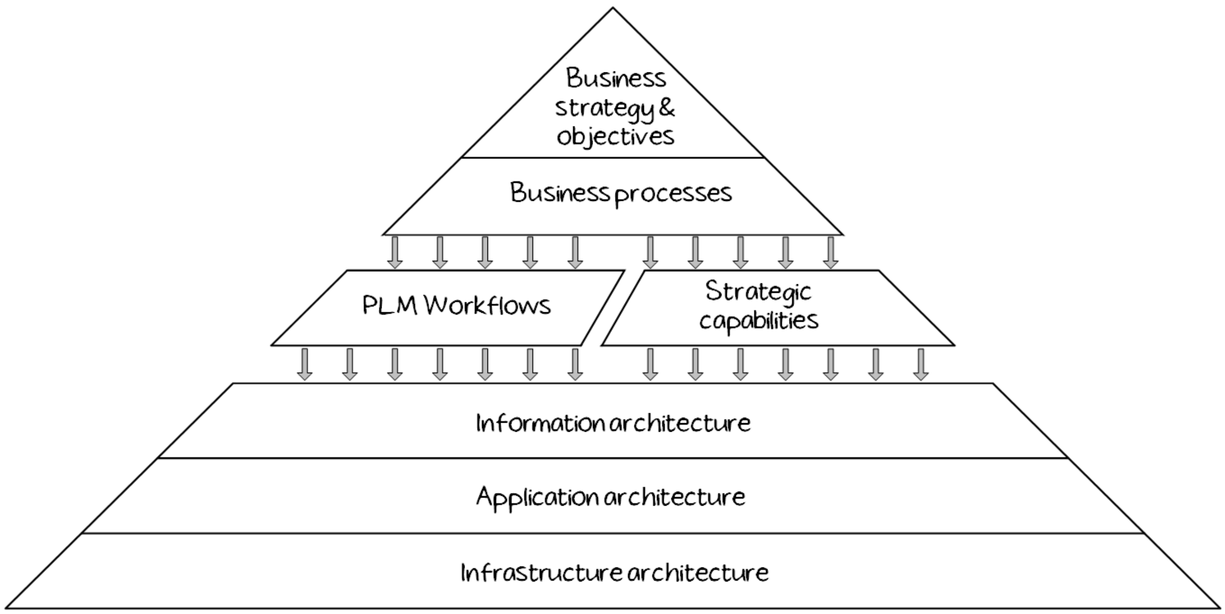


Figure 2-16: PLM Architecture adapted from (Zimmerman, 2008)

The information architecture describes how the information models and their relations support the PLM workflows and enable the strategic capabilities.

The application architecture assigns which tasks are to be performed by which applications.

The infrastructure architecture describes which hardware contains which software e.g. making sure there are enough processing power and storage capabilities for the layers above.

2.5.3 KBE and PLM

From a KBE perspective, PLM can be perceived as the environment in which KBE acts. The perspective that the PLM environment, regardless of its exact constitution, manages all of the product lifecycle-related information to integrate people, processes, information and business systems implies the following kinds of interfaces towards KBE:

- **Information access**
  Interfaces needed for a KBE application to access sources of information and data to read inputs and store/manage outputs.
- **KBE data management**
  Interfaces needed for using PLM to store, version and manage the explicit knowledge applied by the KBE application, e.g. rules, or the application itself.
- **KBE distribution**
  Interfaces needed to execute a KBE application through the user interface of the PLM environment rather than through direct access to the KBE application. The PLM environment is used as a distribution channel for the KBE application.

Taking the perspective of KBE as a process improvement initiative, and considering that PLM constitutes the process and IT environment in product development processes, it is clear that all of these interfaces are activated in different ways. In addition and besides these interfaces, which appeal to the lower three levels of the PLM architecture, there also exist interactions between KBE and PLM in the higher levels of the PLM architecture. Each KBE application has to have a clear role in the architecture and appeal to business objectives and processes either by supporting specific PLM workflows, e.g. automating configuration design or by strategic capabilities e.g. enabling design-for-X methods.

2.5.4 Service-oriented architecture (SOA)

Service-oriented architecture (SOA) is a kind of IT architecture, as the name implies. The most important property of the service-oriented architecture is flexibility, which is enabled by modularization and reusability in the data layer. In a service-oriented architecture the processes/workflows and databases are separated, a principle referred to as the "separation of concerns" (Hurwitz et al., 2007). The principle states that the business processes and the databases are separated by defining the processes as consumers of data and the databases as providers of data; see Figure 2-17. Consequently, a regulation is needed between the provider and the consumer stating what is required and what is delivered. This is referred to as the "service contract". To clarify this setup a commonly used analogy is a restaurant. The menu is the contract and the consumer can see what is offered; the service provider is the chef and the service itself is the delivery of the food according to the contract. As a guest in a restaurant you do not need to know how the food is made, you just need to know what you want to eat. In an SOA, the process does not need to know how the data is accessed, processed, gathered or stored; it just needs to know which data it needs. The needs are met through the service contract stating what the databases can deliver.

The reusability aspect, which enables the flexibility in the architecture, is a result of the implemented contract. When the contract is designed in a way that is not specific to technological solutions possessed by specific IT solutions, it means that other IT solutions which can provide the service according to the service contract can replace existing solutions. This provides flexibility and modularity in the data layer since every service can be said to be performed by a module in the data layer. From the process perspective another kind of flexibility is achieved. With a modularized data layer the processes are not as tied to specific software functions, making it possible to e.g. change the order of activities in the process or assign different roles to improve the process without large adaptations of the software solutions.

Within the area of PLM, a standard called PLM Services 2.0 has been established by the standardization body Object Management Group (OMG, 2007) and developed together with representatives from the German automotive industry. The standard provides the developer of a service-oriented PLM architecture with the contract according to which information is to be communicated. The starting point of the standard consists of the common workflows

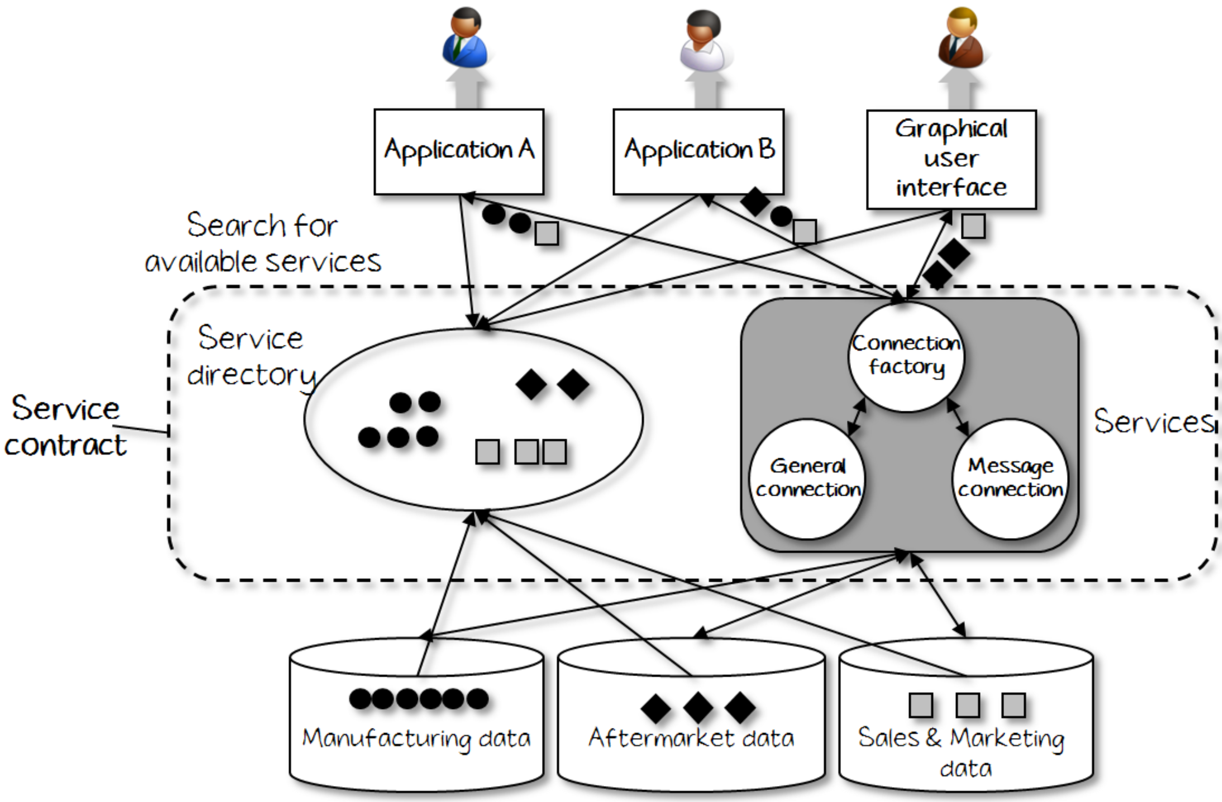


Figure 2-17: Service-oriented Architecture (Ćatić and Andersson, 2008)

encountered in the PLM area, and its aim is to support engineers working with product development. These workflows are generic enough to support certain parts of the product development process but specific enough to be able to implement in software. However, the standard is still in its early phases and still lacks support for basic parts of the product development process, e.g. customer order-driven configuration. Also, it is lacking in documentation and implementation guidelines (Bergsjö et al., 2008).

2.5.5 Conclusions regarding product lifecycle management

The field of enterprise architecture covers in a general sense the need to coherently organize and align the process environment and IT environment. Similarly, PLM aims at aligning product lifecycle processes with corresponding IT support, making the issue of PLM architecture highly relevant from a business perspective. The awareness of PLM architecture has only recently gained attention, due to insights in industrial practice that the issue is not purely IT-related and cannot easily be outsourced (Bergsjö et al., 2008; Zimmerman et al., 2008). The area has high relevance for implementing well-integrated KBE applications since their existence has to be motivated with the processes they support (and those they could support) and the people they interact with. They also have to be positioned in relation to the other IT systems in order not to conflict with architectural principles, e.g. information redundancy or creation of unnecessary interfaces.

The concept of service orientation can make PLM architecture more flexible and easier to manage due to the modularity and exchangeability of the underlying database layer, which is more easily rearranged any time the processes are rearranged or develop different information needs. A service-oriented architecture thus promotes reuse rather than addition of information and IT applications in the PLM architecture.

If translated to the case of KBE implementation, the service-oriented architecture addresses IT barriers, such as a variety of APIs, through neutral information exchange (but not necessarily standard information models). This enables the possibility to modularize KBE applications, each developed with a small effort and limited scope by one or a few engineers, and integrate them to create a bigger whole. An example is the integration of several KBE applications, each designing components from different technology domains, to a product system-level KBE application that designs a mechatronic system. In addition, it makes the implementation easier and more cost-efficient since data can be accessed directly where it resides instead of having to implement dedicated databases between the original data source and the consuming application. These rather technical abilities have relatively significant influences on the higher levels of the PLM architecture as well. The higher flexibility also enables an increased agility in implementing or changing KBE applications as business objectives and processes change in response to the changing business environment of the company.

2.6 Conclusions and identified research gaps

The literature summarized in the beginning of this chapter suggests that KBE is recognized as a promising technology that can revolutionize the product development process. Despite its recognized potential, maturing IT solutions and published demonstrators with documented effects, it has not yet experienced widespread utilization in product development practice.

In Chapter 1 a set of issues with KBE was highlighted. In regard to the issue of “isolated islands” seen from a process and IT perspective, the reviewed literature regarding KBE development methodologies and KBE applications shows a lack of explicit regard for the constraints posed by the PLM architecture in KBE development and implementation, revealing a research gap. This research gap is primarily related to Research Goal A regarding the development and implementation of well integrated and reusable KBE applications. The areas of enterprise architecture and service-oriented architecture indicate that there exist methods and solutions which can enable an explicit regard for the constraints of the PLM architecture into KBE development and implementation and benefit Research Goal A.

With respect to the issues related to the identification of potential for KBE applications, the reviewed literature shows that there is a need for methods of how to identify potential for KBE applications, most clearly observed by Bachrach (1997). The contributions by Sunnersjö (1994) and Pugh (1996) give some guidance in identifying potential for KBE based on certain characteristics of the product and the process. The methodological frameworks of KLIC (Guida and Tasso, 1995) and CommonKADS (Schreiber et al., 2000) make an effort to describe “the typical kind of task” that is suitable for KBE, which reflects a process-oriented perspective adopted in identifying potential for KBE. At the same time, the reviewed literature describing implemented KBE applications provides little detail regarding how the potential for the KBE application was identified. Instead the contributions reflect a technology-driven push approach. Adopting a knowledge management perspective on KBE can give additional support for identifying potential for KBE applications, and there is a research gap regarding the adoption of this perspective in KBE development and implementation. The addressing of this research gap is related to Research Goal B.

It is concluded that the lack of knowledge management perspectives on KBE is a reflection of a general lack of an explicit approach to knowledge management in product development. Contributions such as those of Ullman (2003), McMahon et al. (2004), Ahmed (2005), Wallace et al. (2005), Fu et al. (2006) and Revilla et al. (2010) indicate that there is high relevance in applying models and theories from knowledge management in the context of product development. The Lean Product Development paradigm claims that the main reason behind the successful cases, which constitute the main empirical base for the framework, is an explicitly formulated knowledge management strategy (Ward, 2007; Kennedy et al., 2008). Little support is provided, however, for how an organization should go about formulating an explicit knowledge management strategy other than simply copying Toyota’s. Therefore there is a research gap regarding methodological support for formulating explicit knowledge management strategies in product development. This research gap addresses Research Goal C.

"I am not always able to tell the whole truth,
but I try to tell nothing but the truth."
[Walter Vincenti]

3 Scientific approach

In this chapter the approach for the conducted research is explained in more detail. The purpose is to describe the empirical setting of the research project in order to explain the actions taken, in terms of how and why subjects were chosen for study and methods applied, so as to provide transparency in the research. The first section, however, introduces the research questions along with the rationale explaining how they were derived from the research goals and the identified research gaps. The following section describes the empirical background in terms of the research project setup and the research process, the overall research approach and relations between the undertaken research activities. Finally, a detailed description of each study in terms of purpose, subject studied and methods applied is provided in a paper-by-paper fashion.

3.1 Research questions

In order to be able to discuss the adopted research approach in the coming sections, the research questions are formulated in this section. According to Maxwell's (2005) framework for qualitative research, illustrated in Figure 3-4, the research questions should be rooted in both research goals (described in Section 1.4) and identified research gaps (described in Section 2.6). Now that the reader has been informed about both of these in the previous chapters, the research questions are formulated along with a motivation for each question which explains how it relates to both the research goals and the research gaps.

Research question 1:

How can an explicit regard for the constraints posed by the PLM architecture in the development of KBE applications contribute to a wider utilization of KBE in product development?

This research question is directly related to Research Goal A which aims at identifying ways for developing and implementing KBE applications that are well integrated with the product development process. The question also relates to one of the research gaps, identified in Section 2.6, which states that the constraints of the PLM architecture have not been explicitly considered in the reviewed KBE methodologies and applications. Research conducted in the area of enterprise architecture (Zachman, 1987) states that any IT application development that does not consider the different aspects of enterprise architecture is likely to produce IT applications that are poorly integrated with the enterprise. The question therefore aims at investigating whether and how an explicit regard for the constraints posed by the PLM architecture in the development of KBE applications can contribute to a wider utilization of KBE in product development processes.

Research question 2:

How can the concept of a service oriented PLM architecture enable an explicit regard for the PLM architecture in the implementation of KBE applications?

This research question reflects the first part of Research Goal A which states an ambition to propose and evaluate concepts and solutions that can enable an explicit regard for the PLM architecture. The question thus also relates to the research gap concerning the lack of an explicit regard for the PLM architecture in KBE development. It is, however, also influenced by the indications from literature that a service oriented architecture is a promising solution for resolving some of the major needs related to IT and process integration in product lifecycle management (Burr et al., 2005; Bergsjö et al., 2006).

Research question 3:

How can explicit knowledge management strategies in product development be formulated and methodologically supported?

This research question refers to Research Goal B whose aim is to propose a framework for knowledge management in product development. As mentioned in Section 2.6 there is a research gap regarding a general lack of frameworks for knowledge management in the context of product development. The research question aims at identifying which elements are important to consider and how models and methods from the area of knowledge management can be applied to support the establishment of a knowledge management strategy in product development.

Research question 4:

Which attributes characterize a knowledge management strategy in product development for which KBE is feasible as methodological support?

This research question reflects Research Goal C which aims at identifying potential for KBE applications. The focus of the research question is to investigate which attributes, in a particular knowledge management strategy, indicate high potential for KBE applications. The perspective adopted is that a particular strategy defines the needs for knowledge management of a particular organization, and that KBE may or may not be a possible solution depending on the characteristics of the knowledge management strategy itself. The question relates to the research gap in the studied literature, and practice regarding that potential for KBE applications is identified by defining "typical" tasks and looking for such tasks in a particular organization.

3.2 Research model and empirical setting

The research that serves as the basis for this thesis has mainly been conducted in two research projects which have been set up as joint ventures together with an industrial research partner.

This way of conducting research is strongly influenced by the research model that is described in more detail in Section 3.2.1 below, called the “Wingquist research model”, which is applied in the academic department that has hosted this research project. This particular setup along with the particular industrial partner, described in more detail in Section 3.2.2, has had a large effect on the way the research has been conducted. The research process is detailed in chronological order in Section 3.2.3.

3.2.1 Research model

The model that guides the research in the group where this project has been executed is illustrated in Figure 3-1, and assumes that any research idea has to be based on both a research challenge and an industrial need. This reflects the fact that research in the area of product development aims at practical application of the results and there is a need for any research idea to be well founded both in industry and academia. The research idea is most commonly embodied in some kind of demonstrator with the purpose of demonstrating a technology or a new method and tool whose aim is to affect the product development practice. As a whole, this model also reflects the general flow of the design research methodology (DRM) proposed by Blessing and Chakrabarti (2009) illustrated in Figure 3-2.

3.2.2 Research setup

The research project was managed and executed jointly together with Global Corp., as illustrated in Figure 3-3. This is where the research ideas were formulated and executed in the form of studies, demonstrators and implementations. The reference group consisted of representatives from different divisions of Global Corp. and from Chalmers. Their backgrounds were mainly related to IT and process support for product development in their respective divisions. The group met on a monthly basis to evaluate proposed research ideas based on industrial needs and

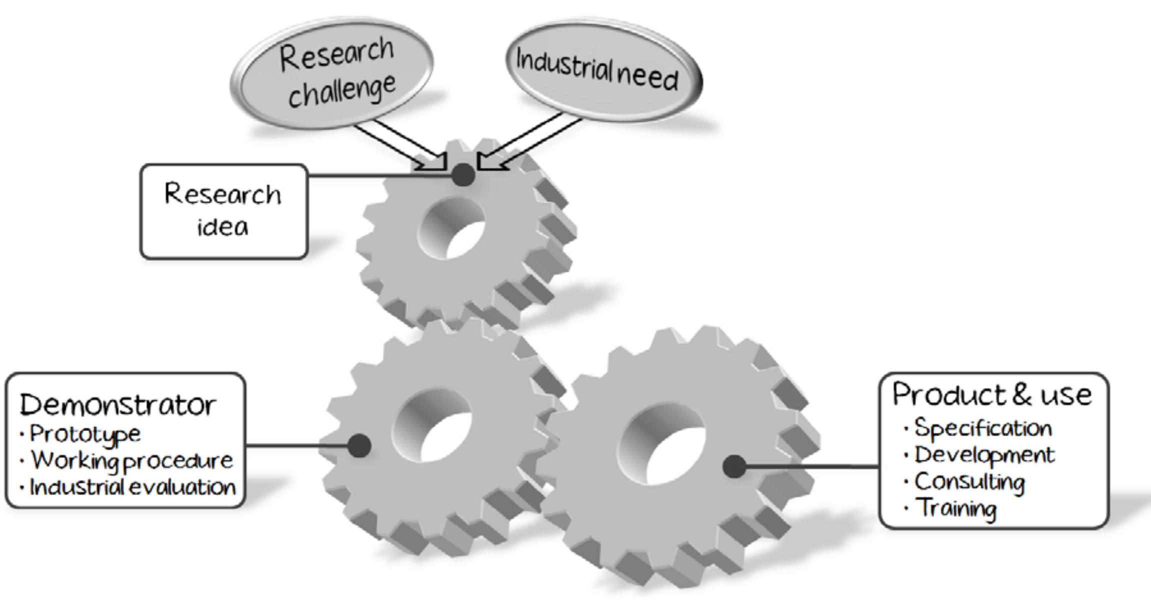


Figure 3-1: The Wingquist research model

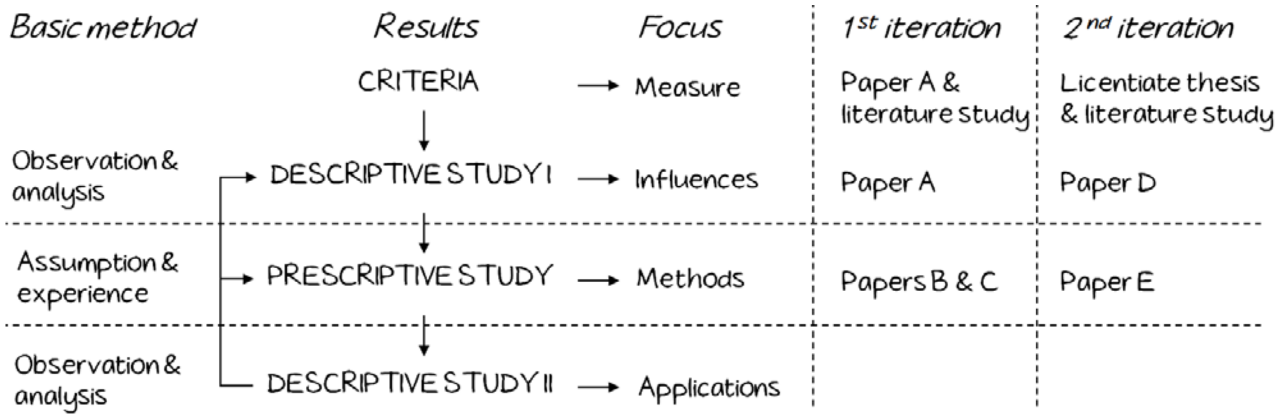


Figure 3-2: The appended papers put in relation to the DRM framework

research challenges, as illustrated in Figure 3-1. They also identified and ensured access to key individuals that could support proposed studies and demonstrators in different ways, e.g. line managers who provided engineers for interviews, or process and IT owners who could support the development of demonstrators. Furthermore, the group supported the execution of studies and demonstrators by resolving issues encountered along the way. In addition, the research project had a group of supporters who, for different reasons, supported the project in informal ways. They proved to be of critical importance for some of the research activities, as described in Section 3.3. The primary subjects of study have been the Driveline, Aerospace and R&I divisions. These divisions have acted as the sources of empirical data for different issues regarding product development practice that have been studied, and as evaluators of results and proposed solutions.

3.2.3 Research process and design

The chronological order of the publications and reviewed literature topics is illustrated in Figure 3-5, and the abbreviations used in Figure 3-6 relate to the literature topics in Figure 3-5. According to Maxwell (2005) every research effort has an inherent design which governs

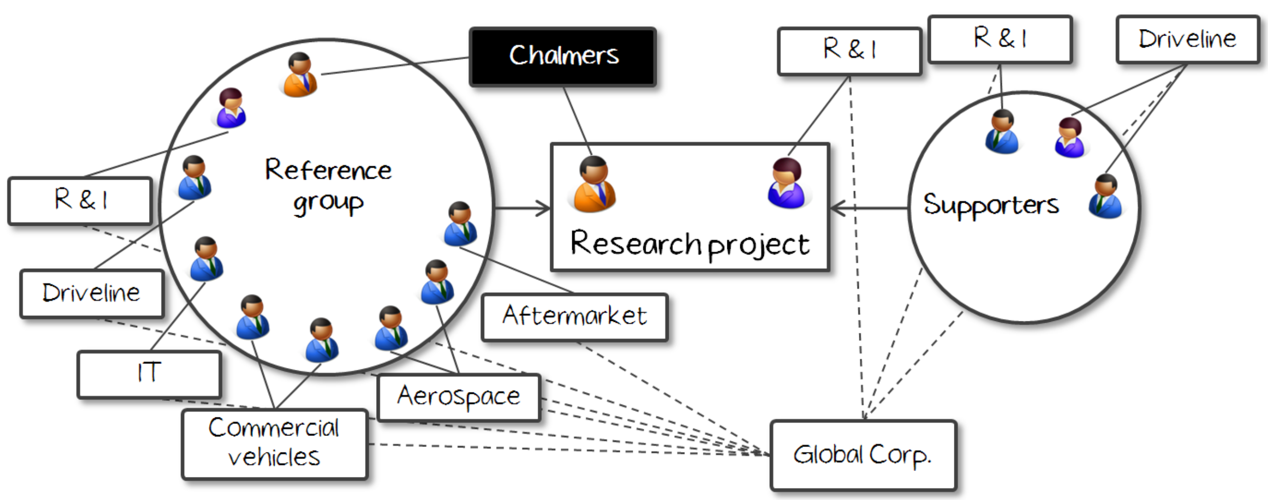


Figure 3-3: Research project setup

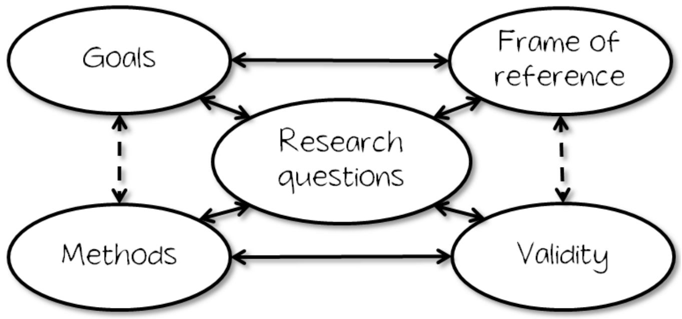


Figure 3-4: Maxwell's (2005) research design framework

its execution and subsequent validity. Maxwell states that there are five main components which constitute the architecture of each research design and affect each other explicitly and implicitly, as illustrated in Figure 3-4. The first research activity was initiated with the formulation of Research Goal A (as indicated in Figure 3-6) which governed the mapping of the state of the art and state of the practice for KBE development and implementation. This resulted in the formulation of Research Questions 1 and 2 which were subsequently addressed in Paper B and Paper C. Using the Design Research Methodology (DRM) (Blessing and Chakrabarti, 2009), Paper A constitutes a Descriptive Study I, as illustrated in Figure 3-2, which describes the issues related to the integration of KBE with product development. Papers B and C are Prescriptive Studies that describe how those issues can be addressed with an explicit regard for the constraints of the PLM architecture in KBE development and implementation through two demonstrators. These three papers constituted the main foundation for the licentiate thesis, which was concluded with the need to study KBE from a knowledge management perspective and led to the formulation of Research Goals B and C which in turn have been formulated as Research Questions 3 and 4.

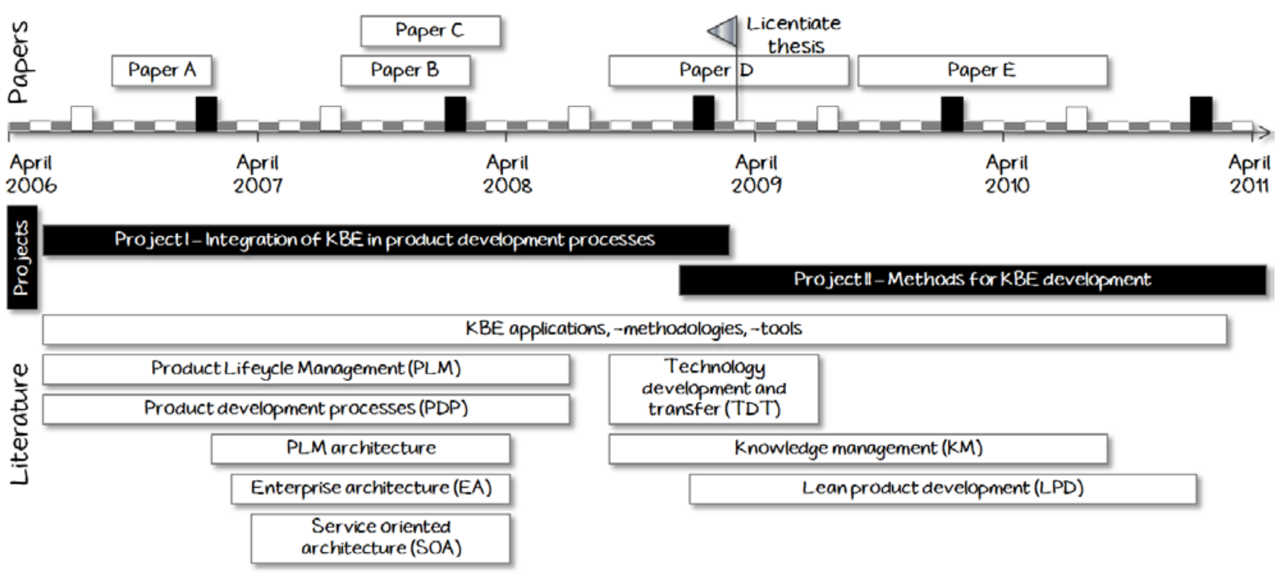


Figure 3-5: Publications and reviewed literature topics, in a chronological order

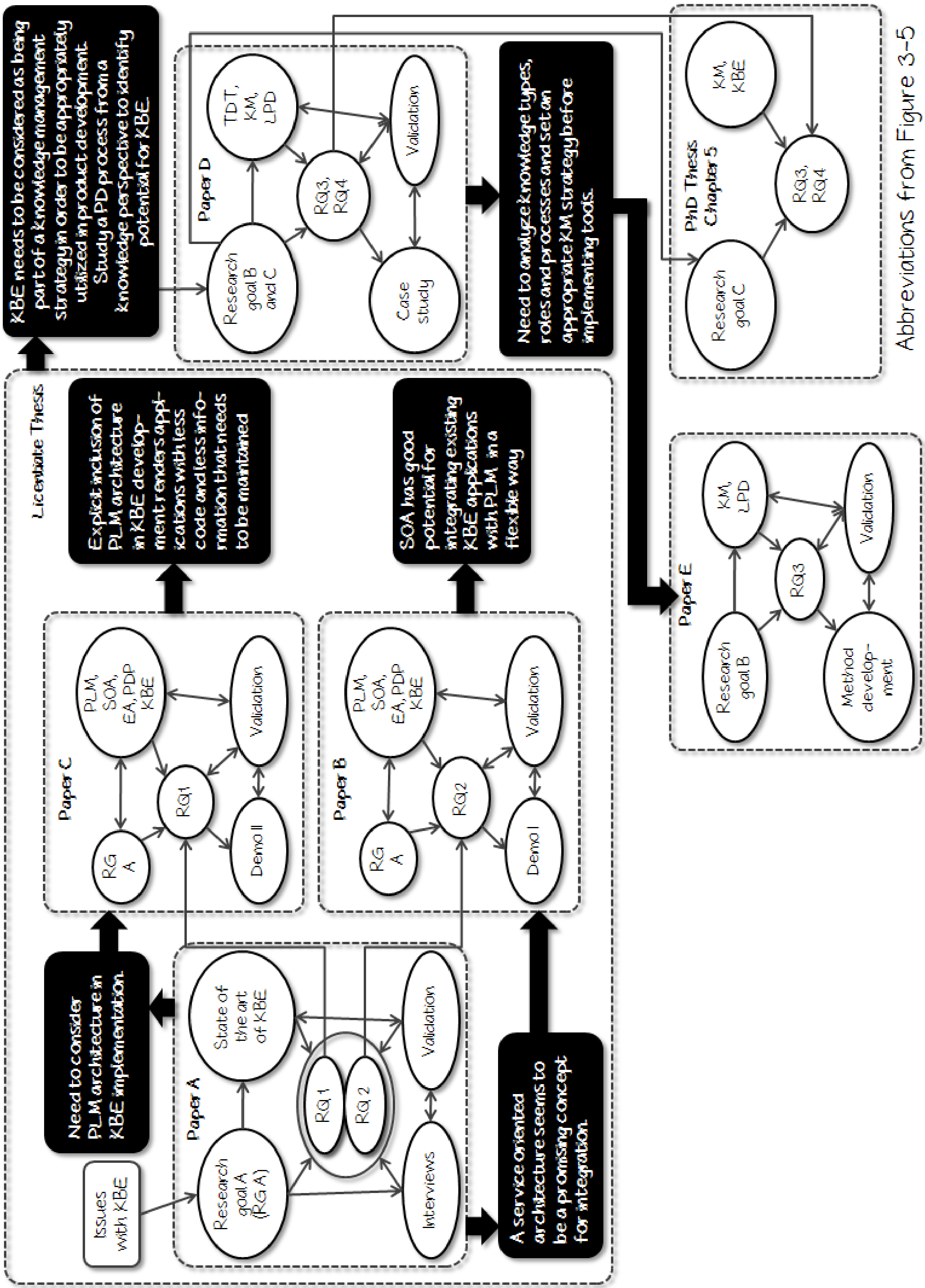


Figure 3-6: The research process visualized using Maxwell's research design framework for each paper and with relations between the publications

In Paper D, which is a new Descriptive Study I using DRM terminology (see Figure 3-2), a case study of a product development project from a knowledge management perspective was performed. The main purpose was to study how knowledge management is approached in product development practice and to investigate whether potential for KBE can be identified using this approach. The results indicated a need for research regarding how to approach the concept of knowledge management in product development, which was the main reason behind the research conducted in Paper E. Since Research Goal C and Research Question 4 were left unaddressed, Chapter 5 of this thesis makes an attempt to address these with a framework that is based on results and experiences obtained during the course of the two research projects. The framework constitutes a second Prescriptive Study according to DRM terminology, and has not yet been validated in product development practice.

3.2.4 Validation approach

The validation approach cannot be said to have consisted of one kind of strategy. The different activities and methods studied have made it necessary to address validation in different ways. Maxwell (2005) proposes eight strategies for continuously addressing validity during the course of qualitative research, and these are referred to for each of the studies as well as for the research project as a whole:

1. **Intensive long-term involvement**
   Long-term participant observation yields a more complete view of the subject under study from more perspectives along with a better understanding of its context.
2. **"Rich data"**
   Detailed and varied data that describe a fuller and more revealing picture of the studied subject. Rich data also provide the reader with higher transparency regarding conclusions.
3. **Respondent validation**
   Soliciting feedback regarding the collected data from the studied subjects.
4. **Intervention**
   A conscious and experimental manipulation of the studied subject and a close study of the reaction.
5. **Searching for discrepant evidence and negative cases**
   The identification and analysis of cases which do not comply with the conclusions of the researcher provide a good source for either improving the conclusions or strengthening them when other causes of the discrepancy are found.
6. **Triangulation**
   Data collection from various sources and settings using diverse methods to either confirm or identify discrepancies.
7. **Quasi-Statistics**
   Statements relating to quantitative appreciations of observed phenomena such as 'rare', 'often', 'typical' should be backed up by numerical results derived from the data.

8. **Comparison**
   Comparing e.g. cases to other cases or results to existing results provides the possibility to identify aspects which either complement or strengthen the conclusions.

These strategies are however primarily related to the process whose purpose is to provide transparency, and also to minimize the risk of bias on the part of either the researcher or the research subject. Another aspect of validity is primarily related to the research results and consists of the transferability (the extent to which the results can be transferred to other similar cases) and the generalizability of the results (the extent to which the results are valid beyond the specifics of a certain case). These aspects are discussed in Section 6.2 after the research results have been presented.

3.3 Adopted research approach

The logical thread between the papers is illustrated in Figure 3-6 indicating how the conclusions from each paper have affected the direction of the next paper. For each of the papers the choice of methods has been related to the goals and research questions at the time of execution, but these have been refined in between the papers. The research questions and goals stated in this thesis are the ones that form a unity together with the appended papers and "tell a complete story". In addition, the subsections below describe how the previously mentioned validation strategies have been applied in the research project as a whole and in the activities related to each paper, and also how these strategies have affected the research process.

3.3.1 Review of KBE literature

In the literature study, publications dealing with KBE methodologies were studied along with KBE applications developed and evaluated or implemented in an industrial setting (summarized in Appendix A). The literature search was performed primarily through scientific journals in the field of design science and proceedings from the largest conferences in design science, searching for the terms "knowledge based engineering", "KBE", "knowledge management" and "design automation". Most of the KBE applications were found in this way. Two of the applications (numbers 13 and 23) were identified by using personal contacts which later referred me to the publications. In addition, websites of scientific journals in the fields of knowledge-based systems, expert systems and artificial intelligence were scanned for titles and abstracts indicating that an application has been developed for an industrial process and demonstrated/evaluated in its real-life context. No such applications were found. The publications deal with what is referred to in Section 1.5 as the "inner workings" of KBE, such as problem-solving algorithms applied to engineering problems but lacking connections to a real engineering product development process deploying IT solutions that can be found under the PLM umbrella. An example of an application that did not meet the criteria of real-life context can be found in (Wojciech, 2007). Publications related to KBE methodologies summarized in Section 2.1.1 were identified through references from the publications of KBE applications.

3.3.2 *Paper A – Towards integration of KBE and PLM*

The primary aim of Paper A was to identify different reasons why KBE is poorly integrated and potential solutions for addressing those reasons. The primary purpose was to describe the state of the art and practice of KBE development and implementation so as to set the agenda for the research project. The secondary purpose was to orient myself in the area of KBE, of which I had little previous experience.

Since the activities were initiated with a research goal and it was a descriptive study, the research methods applied to collect data were the previously described literature study, to map the state of the art, and a survey, to map the state of the practice, of KBE development and implementation. The industries dominating the publications are the automotive and aerospace industries, but there are examples from other types of manufacturing industries. Four aspects of each application were studied; what it does, how it is motivated, what its purpose is and how it interacts with its surrounding process and IT environment.

The survey was planned by myself and iterated with the reference group who gave feedback on its focus. The interviewees consisted of eight experts who had developed and implemented KBE application and two users of some of those applications. Six were from Global Corp. (four of them were in the reference group) and two were from a mid-size company that develops and manufactures submersible water pumps (whom I had found using my personal networks). No particular method was applied for selecting interviewees, but personal networks were used which is acceptable because the survey did not require a randomized sample (Williamson, 2002). The data were collected using semi-structured interviews lasting 1-2 hours. Each expert was asked to explain what the KBE application does, how the idea for it came up, how it was developed and how it interacts with its surrounding processes and IT. The users were mainly asked to give their points of view on the role of the KBE application in the process and how they interact with it. The collected data were analyzed by looking for patterns of common and differing elements in the publications and the interview answers regarding the way the KBE application is integrated and utilized in the process it supports.

The results from the literature study and the survey were presented to the reference group. A similar presentation was held at the mid-size company in the form of a workshop. The reference group confirmed that the lack of explicit regard for the PLM architecture in KBE development and implementation reflected an industrial need and that the concept of a service-oriented architecture was worth investigating through the development of a demonstrator.

Being a descriptive study that is based on interviews and a literature study for data collection, the validation strategies of respondent validation, triangulation and comparison (Maxwell, 2005) were used. Respondent validation was used for two reasons; one was to make sure that I had the same understanding as the interviewees regarding the specifics of the KBE applications in question, and the second was to allow the respondents to give their feedback on the conclusions about the lack of regard for constraints of the PLM architecture in KBE development and implementation. All of the experts confirmed the results and the need for more research during the two workshops in which the results were presented and discussed.

Triangulation was used in two ways and for two purposes. Firstly the interviews with the KBE users were used to see if they had a similar view as the experts on how the KBE applications support the process, which they did. Secondly the experts from the mid-size company were used to compare the results from Global Corp. in order to identify potential biases related to company size or line of business; no such biases were found.

Comparison was finally used to compare the KBE applications between each other. An interesting result was that two of the applications (numbers 13 and 23 in Appendix A) had a significantly higher number of users and were more integrated with a higher number of processes. Coincidentally, the design of these applications also had a more explicit inclusion of the PLM architecture in terms of integration with data sources and end user IT systems, and they were explicitly included in process descriptions.

3.3.3 Paper B – Implementing a service-oriented PLM architecture focusing on support for engineering change management

The idea to test the concept of service-oriented architecture (as reflected in RQ2) set the stage for choosing the development of a demonstrator as the research method, especially since demonstrators in different forms are an effective way of making theoretical concepts concrete and easier to evaluate in practice (Williamson, 2002). One of the supporters of the research project helped us arrange for the setup of a master thesis project (Persson and Stiborg, 2008) and assigned an expert on service-oriented architecture as the industrial supervisor and myself and my colleague Dag Bergsjö as academic supervisors. In parallel the reference group representative from the Driveline division identified the engineering change management process as suitable to study, especially since one of the KBE applications studied in Paper A (not included in Appendix A due to confidentiality reasons) already supported that process in a not so integrated way. The research idea was to enable a better support of this process using existing elements but integrating them more by using a service-oriented architecture based on the standard PLM Services 2.0 (OMG, 2007).

The data collection consisted of three interviews with three separate individuals to

1. document the steps in the process and IT systems used (together with the process owner),
2. identify issues and obtain data used in the process (together with a group manager)
3. obtain, understand and adapt the KBE application used (together with the expert who developed it).

This gave us a deeper understanding of the industrial issues and needs. The data collection consisted of daily discussions with the two students and the expert which were documented in weekly plans detailing issues encountered and how they were resolved.

The demonstrator was evaluated through presentation and evaluation to the initial process owner and group manager, the reference group, a process and IT committee with experts from all of the Global Corp., a committee of top-level product development managers, and finally to the manager of an industrial project aimed at integrating two large databases with a service-oriented architecture. In addition, the demonstrator was presented at an internal technology fair where approximately 100 line managers from different divisions of Global Corp. attended. All these fora provided us with different kinds of feedback from their perspectives regarding the relevance of the demonstrator.

The validation strategies applied in Paper B had four different purposes. One was to ensure that the engineering change management process was correctly understood and described. The second was to ensure that the demonstrator indeed addressed the issues identified in the process. The third was to address the validity of the concept of service-oriented architecture, and the fourth was to validate the results in relation to Research Question 2. For the first and second purposes, the strategy of respondent validation and triangulation was used, when the interviewed individuals were asked to reflect on the description of the process and the issues in the process. Later the same individuals were asked to provide their feedback regarding the demonstrator, which was positive. Triangulation was the main reason behind choosing a process owner, a KBE developer and a line manager, to see whether their different perspectives provide a similar view of the process and of the issues in the process.

To address the validation of the SOA concept, the strategies of triangulation and comparison were used. Triangulation was mainly realized through the activity of presenting to the previously mentioned fora and asking for feedback from individuals with different backgrounds. Most of the individuals involved with different kinds of business operations confirmed that there is a need to integrate KBE and the need to integrate information in heterogeneous IT environments. Individuals with experience of process and IT management confirmed the validity of SOA at a conceptual level but were skeptical to its scalability in an industrial environment. There existed also a possibility to use comparison as a validation strategy since two other prototypes using the SOA approach had been recently developed within Global Corp. with positive outcomes. A comparison showed that one of the prototypes used the same standard as our demonstrator with similar experiences and issues as described in Paper B. The other prototype used another standard called OASIS and did not experience these issues. The OASIS prototype was rolled out in production in a real product development project and showed that the SOA concept is scalable in an industrial environment.

Finally the demonstrator was compared to the previously analyzed KBE applications from Paper A in order to validate it in relation to Research Question 2. The main strategy adopted was comparison, where the KBE applications were compared as to whether or not they contain hard-coded and duplicated data (and thus introduce redundancy) and whether or not they contain hard-coded interfaces (and how many) towards other systems.

3.3.4 Paper C – Manufacturing experience in a design context enabled by a service-oriented PLM architecture

Inspired by the demonstrator development in Paper B, members of the reference group from the Aerospace division identified a process that is concerned with reuse of manufacturing knowledge in product development and had high potential for automation due to its repetitiveness and time consumption. The representatives proposed the development of a KBE prototype as a demonstrator in which the effects of an explicit regard for constraints of the PLM architecture on the development of a new KBE application could be studied. The development of the KBE prototype was set up as a joint venture together with Petter Andersson, an industrial PhD student from Global Corp. Aerospace division and Luleå University of Technology.

In order to document the process along with the pieces of information used and the IT systems where they originate, interviews were held with three designers. Several basic concepts for the KBE application were proposed and iterated with a KBE expert (also a member of the reference group from Aerospace) to agree on a final concept. Petter Andersson performed the software programming of the final KBE application and it was evaluated in a workshop where it was presented to the intended users, who gave their feedback. It should be noted that the KBE application, though developed and positively evaluated, was never launched in production use. An attempt was made but the discussion with owners of specific IT systems never rendered any fruitful solutions regarding access to real information from their systems. These discussions, however, did provide some useful insights regarding the implementation of the concept of SOA in PLM.

Similarly to Paper B the main validation strategy adopted in Paper C, for validating that we had a good understanding of the process and the IT environment involved with the reuse of manufacturing knowledge in product design, was respondent validation and triangulation. In the triangulation the respondents' answers were compared with each other and also with a formal process document to establish the flow of the process, the information and data used and the IT systems where those originate. Once we had an established map it was presented to the interviewed designers to see if they agreed on it, which they did.

Through the previous activities the tasks executed by the prototype were validated from the perspective of a designer who wishes to access and reuse manufacturing knowledge. In order to validate the demonstrator in relation to Research Question 1, a comparison was made with three of the KBE applications from Paper A whose source code we had access to. The comparison showed that our demonstrator had fewer code lines related to hard-coded data and hard-coded interfaces.

3.3.5 Paper D – Requirements management when introducing new mechatronic subsystems – managing the knowledge gaps

To address Research Questions 3 and 4, which require answers with prescriptive elements, a descriptive study had to be undertaken first to investigate the status of knowledge and knowledge management in product development, which was the starting point for Paper D. The research idea was to study a particular product development process and project from a knowledge perspective. The industrial need is based on the recognition that truly effective product development rests not only on effective data and information management but also on effective knowledge management. The research challenge lies in applying the theories and models from the field of knowledge management to the field of product development.

The choice of method was discussed in the reference group. The choice fell on an exploratory case study which, according to Yin (2003), is defined as "an empirical inquiry that investigates a contemporary phenomenon within its real-life context especially when the boundaries between phenomenon and context are not clearly evident". The reason for choosing a case study is that the studied phenomenon is knowledge and its management and the only way it manifests itself is by supporting decisions and activities in a process. There is no clear boundary between the knowledge and the effects resulting from its application. In addition, since most product development is carried out in the form of projects, choosing a particular project is a practical way of defining the boundaries of the study.

The supporter of the project from the R&I division helped us in selecting appropriate cases to study. Three different cases were discussed. All were recently finalized projects in the Driveline division. The selected case was a project in which a new kind of driveline containing a completely new subsystem was developed. The reason for choosing this project was that it was particularly "knowledge-intensive". It contained technology and knowledge transferred from the R&I to the Driveline division, with new individuals who had to develop new product and process knowledge but also to reuse a lot of existing product and process knowledge from the established departments. The focus of the study was on identifying how knowledge was currently managed in the organization, what kinds of new and existing knowledge there were, how the new and existing knowledge is managed and how existing knowledge was utilized in the process, in order to investigate whether there were initial knowledge gaps in the project and if these knowledge gaps later affected the project. An additional purpose was to investigate whether there exists potential for KBE applications in the studied process, in order to see if there is a connection between certain attributes in how the knowledge is managed and potential for KBE applications.

The data collection was done mainly through interviews carried out by myself and the research project member from the R&I division. A total of 15 individuals were interviewed during a period of two months. Two of them were contacted by our supporter from R&I and both became our supporters from the Driveline division. One of them was the manager for the department who designed the new subsystem and the other one was a manager for a group of project managers some of whom were involved with the studied project. These two supporters in turn selected the engineers, based on the case study focus, whom we later interviewed. The interviews lasted approximately two hours and were semi-structured. Each interviewee was asked to reflect upon issues that occurred in the project and how these could have been avoided, and also how they were resolved. They were also asked to reflect upon important decisions and how these decisions were enabled. In addition they were asked to reflect upon individuals they consider knowledgeable, which knowledge they have and how they relate to these, IT tools and systems that are considered to manage knowledge (and which knowledge), and their role and knowledge gaps encountered during the process, along with which kinds of knowledge they considered this to be. Finally they were asked to reflect on whether there exists potential for one or several KBE applications to either support certain activities or manage certain knowledge.

In the analysis of the data, the interviews were transcribed and patterns were identified. Comments were related to different topics such as issues, decisions and types of knowledge created/reused/needed related to different aspects of the development process such as requirements management, interface management, requirements verification and testing, organization and suppliers. The results from the study were presented and discussed in three workshops covering (1) requirements management, verification and testing, (2) interface management and (3) organization and suppliers. The workshops were used as a way to get feedback regarding the findings and discuss possible solutions. The participants for the workshops were selected based on their role and which topics they talked about during interviews, so that a good match was found between participants and workshop topic.

The final results from the study were discussed with the reference group and the project supporters, to decide upon the continuing activities regarding the development of demonstrators which address the identified issues either related to knowledge management or KBE development. A decision was reached that a method for managing the knowledge which currently is not managed but is of high importance for the process should be the next step for the research project, in order to be able to address Research Question 3. The addressing of Research Question 4 did not render any fruitful results in this study, so an attempt to address it is made in the end of Chapter 5 of this thesis.

The validation strategies related to the case study are primarily concerned with making sure that a good understanding was obtained of what happened in the studied project, and that the subsequent conclusions regarding identified issues were in harmony with the perspectives of the respondents. The strategies adopted were respondent validation in which all of the respondents were provided with summaries of their interviews; no changes were requested. The respondents who were chosen for the workshops were also given a chance to elaborate a richer feedback. The final report was also e-mailed to all of the respondents with no requests for changes. Triangulation was used in two ways. One was between the responses obtained by the line organization members and the members of the project team (as illustrated in Figure 1 in Paper D), specifically regarding project-related events such as specific schedule slips. Different pictures were conveyed by the two groups of respondents, which revealed a knowledge gap regarding process knowledge among the line organization members. Another type of triangulation was made between the respondents' replies related to issues regarding requirements management, requirements specification documents and the IT support for requirements management, which confirmed the stated issues. In order to address the validity of the results obtained in the case study, the strategy of rich data was used in the writing of the case study report, which is an internal report at Global Corp. Elements of the report are reproduced in Paper D with the exception of confidential parts. Finally, comparison was used in suitable aspects with literature such as supplier collaboration or technology transfer. The comparison confirmed the results in the different aspects, as similar results have been reported as referred to in the recommendations of Paper D.

3.3.6 Paper E – Knowledge management in mechatronic product development: effective method for creating engineering checklists

Paper E was a prescriptive study with the aim to evaluate a methodological solution. Even though structured methods are proposed for prescriptive studies in the DRM framework, the approach adopted for the development of the method was rather pragmatic. The decision to use engineering checklists as the knowledge management method was preceded with a consideration of several alternative methods such as writing lessons learned, establishing "yellow pages", and establishing a database of guidelines and best practices. The alternatives were, however, evaluated with two line managers and five process owners and they were either already tested (lessons learned and guidelines/best practices) or found unsuitable due to corporate culture (yellow pages). Engineering checklists as a method for managing interface-related knowledge were found worth pursuing but needed to be complemented with a method that supports their creation.

In the evaluation phase, individuals were selected partly from the originally studied department with the help of the department and partly from the R&I division with the help of the research project member from R&I. The second sample of individuals (from R&I) was chosen for triangulation purposes, and the selection criteria was that their operations should mainly deal with product development of a subsystem in which interface management is of high importance. The individuals were provided with the proposed method in the form of a macro application implemented in a commercial spreadsheet software and a short guide. After making the first checklist the participants were interviewed for approximately 30 minutes in structured interviews, to provide their feedback regarding the usability of the method and the usefulness of the results generated by the method.

The two validation strategies of interest for Paper E have been respondent validation and triangulation. The purpose of the respondent validation was to obtain feedback regarding the proposed method, in terms of whether it addresses the management of knowledge they consider as important and whether it does so in a useful way. One of the issues encountered was that, of the five engineers from the Driveline division who were asked to evaluate the method, only two replied, both of whom gave positive feedback. The five engineers from the R&I division all replied with positive feedback. The triangulation strategy was applied in two different ways for two different purposes. The first kind of triangulation was that the five engineers from the Driveline division were from the same group that was initially studied but none of them was interviewed in the case study. The purpose of this was to offset the risk that the proposed method focused on resolving issues that were unique for the interviewed engineers. The second kind of triangulation was the selection of the second group of engineers from the R&D division. The purpose of this was to offset the risk that the method was resolving issues unique to the studied group of engineers in the Driveline division.

3.3.7 Overall validation approach in the research project

Besides the validation approaches adopted in regard to each paper, two validation strategies have been continuously applied during the course of the two research projects, namely the intensive long-term involvement and the search for discrepant evidence and negative cases. The involvement with the Driveline division can be considered as long-term since it was initiated in June 2006 and lasted until September 2010 when the last evaluation was carried out for Paper E. This organization has been the main subject under study, which has brought about not only an understanding of its products, processes, methods and tools but also its organizational roles, culture and relations towards other divisions within Global Corp. This has yielded an understanding of what differentiates this particular division from other types of product development organizations, making it easier to make comparisons to other organizations in the discussions regarding the transferability and generalizability of the results. The search for discrepant evidence and negative cases has been performed inherently in the literature studies carried out during the project but more importantly through discussions with other researchers and industrial practitioners who were considered as potential sources of perspectives and experiences that were incompatible with my own conclusions. One such case was the already mentioned skepticism regarding the scalability of the SOA concept due to technical issues related to lacking information models. Our attempts to apply an SOA integration of the KBE prototype developed in Paper C, however, showed that the main issue affecting the scalability of the SOA concept in practice is more related to the fact that an IT strategy has resulted in an IT environment where the variety of chosen standards constitutes the main barrier.

3.4 Summary

Having covered the needs and goals of the research project in the introduction and identified research gaps in the frame of reference, the aim of this section has been to highlight the research approach adopted for addressing the needs and research gaps with research questions which were used to justify the methodological choices. The overall approach reflects the framework for qualitative research design proposed by Maxwell (2005). The adopted approach is mainly qualitative in order to deal with the socio-technical environment of product development processes in which cause and effect relations are complex, and thus a linear research process is not always possible or suitable. In the next chapter the results of the research are presented in the form of summaries of the appended papers.

"The only source of knowledge is experience."
[Albert Einstein]

4 Results

In this chapter the results from the appended papers are summarized. They constitute the main body of theoretical and empirical findings on which this thesis is founded.

4.1 Paper A: Towards integration of KBE and PLM

The purpose of this paper is to analyze KBE from a product development and product lifecycle point of view. The research question guiding the effort concerned a concretization of issues which affect the integration of KBE with its surrounding PLM environment. The aim was to assess potential solutions which address the identified issues.

The results from the study of the published KBE applications (see Appendix A) and those implemented in Global Corp. and Mid-size Corp. show that the poor utilization of existing KBE applications is directly related to the integration of the KBE applications with the PLM systems. It was observed that the implemented KBE applications in the two companies were isolated from the PLM systems, and in the most extreme cases were unheard of by the actors in the processes which they supposedly supported. Most commonly the KBE applications supported the processes implicitly through a human interface (usually the expert who developed it or an expert user who was involved in the development). Two exceptions exhibited a different behavior (Strinning, 1995; Fuxin, 2005). Being more integrated with the PLM systems, they also had a wider utilization in the processes.

To enable the PLM integration, which would enable a better process integration and wider utilization, a service-oriented architecture (SOA) was proposed. In an SOA, KBE applications and other components of the PLM environment are regarded as providers of services to the process. Services can be configuration, calculation, design or more fundamental PDM functions such as provision of information related to an article number or storage and versioning of virtual models. A concept called "knowledge modules" is introduced to denote a set of KBE applications which use each other's services to perform a task which, from the process point of view, has a higher degree of completeness. For example, in the sales process for trucks, a customer might want to know the fuel consumption of his/her particular truck configuration. A knowledge module performing this task (calculation of fuel consumption for a particular truck configuration) would employ a KBE application that configures the truck, then a second to calculate the weight, a third to establish the load profile based on the customer's user scenario, and finally a fourth which uses the weight and the load to calculate the fuel consumption. All of the mentioned applications have high relevance for processes in engineering, but the knowledge module can use them to perform yet another task in the sales process without the need to make a dedicated (and rather complex) KBE application. To enable such a scenario, a highly flexible IT architecture is necessary, which is the reason for proposing an SOA.

The main conclusion from Paper A is that an explicit regard for architectural constraints in KBE development and implementation is rare, which negatively impacts the integration and utilization of KBE in the processes. This conclusion constitutes the main background for Research Question 1 of this thesis. Another conclusion is that already implemented KBE applications support the processes indirectly through a human interface (they are usually only used by the experts who developed them). Finally, it was concluded that an SOA is a promising solution to address the issues identified in both of the previous conclusions. This conclusion constitutes the main background for Research Question 2 of this thesis.

4.2 Paper B: Implementing a service-oriented PLM architecture focusing on support for engineering change management

As SOA was identified as a potential enabler for integrating KBE and PLM, the focus of Paper B is on implementing an SOA in a demonstrator to evaluate this concept. The purpose was to investigate how a service-oriented PLM architecture can be implemented and how it can benefit the integration of KBE and PLM to address Research Question 2. Since the concept of SOA cannot be isolated to the area of KBE alone and embraces the total PLM environment, issues related to PLM architecture are included as a significant part of Paper B.

Prior investigations from Global Corp. show that the lack of explicit regard for the constraints of the PLM architecture is not specific for KBE but is also present in PLM implementation in general (Zimmerman et al., 2008). A common solution to avoid issues related to PLM architecture, promoted by the PLM vendors and initially adopted by some OEMs, was to use a so-called “single-source” database for managing all product-related data and supporting all product development and lifecycle processes. The single-source solution is quite different from the typical previous legacy solutions, which tend to be dispersed and implemented locally in the different engineering departments supporting local processes very well, but resulting in an overall suboptimal process and IT environment (Svensson, 2003). Due to these experiences an SOA solution based on a standard called PLM Services 2.0, which lies between the two extremes of single source and sub-optimized legacy, was chosen for the demonstrator.

The demonstrator supports an engineering change analysis process and is specifically focused on how a change in a certain diesel engine component (the turbo charger) affects the rest of the engine. In the demonstrator, the user creates a change in the specification of the turbo charger and initiates a change request, which is handled as a workflow in a dedicated application which can be considered to be a knowledge module, as described in Paper A. In the workflow, a legacy KBE application is executed to simulate the impact of the changed component on the key properties of the system. The effect of the new properties on other components is obtained by

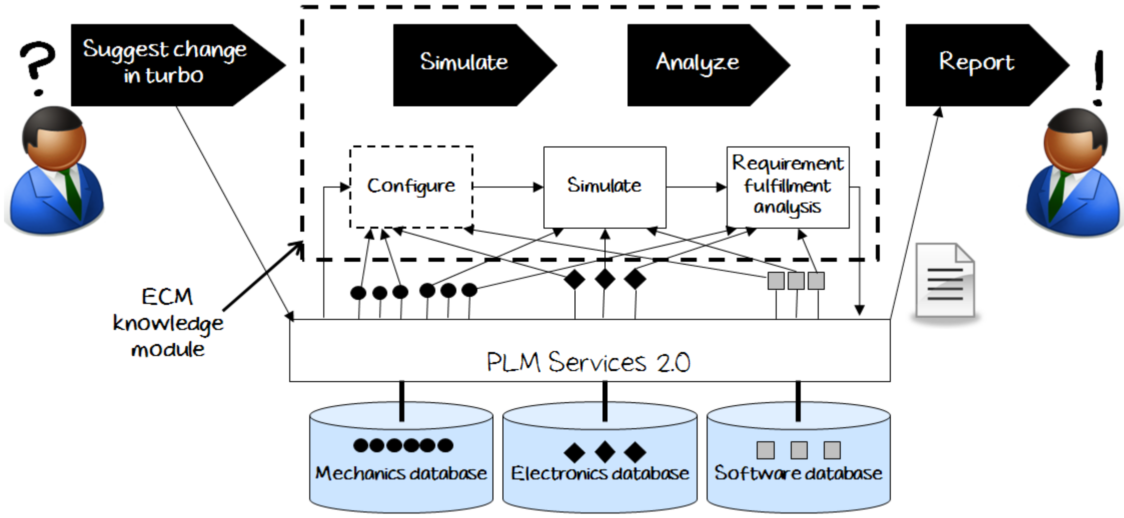


Figure 4-1: The process and IT architecture of the demonstrator in Paper B

comparing the result of the simulation with the specifications for the affected components inside the knowledge module. The component specifications reside in their own databases and are gathered at run time as input to the KBE application. Finally an e-mail is sent back to the change initiator to inform which specifications are no longer satisfied due to the change; see Figure 4-1. The sequencing and execution of the activities performed above the service layer are run by the knowledge module.

The results related to KBE implementation and integration indicate that the SOA concept enabled the existing KBE application to integrate better with the engineering change process it indirectly supported through a human interface earlier. The results also show that the efficiency of the EC process could be increased due to quicker and more extensive feedback regarding effects on other components.

The conclusion from Paper B related to KBE is that a service-oriented architecture can enable an explicit regard for the constraints of the PLM architecture in KBE implementation by allowing the KBE application developer to focus on supporting the process in an optimal way. The fact that issues related to interface development and data access are eliminated (from a technical point of view) means that there is no need to trade off functions in the KBE applications which require data that would be costly to access due to interface development and maintenance. Another conclusion is that the maintenance of the KBE application is facilitated since there is a minimum of interfaces which need to be maintained. It is also concluded that an SOA enables the development of process-oriented knowledge modules which reuse existing KBE applications and product information by reducing the amount of issues related to IT integration.

4.3 Paper C: Manufacturing experience in a design context enabled by a service-oriented PLM architecture

In Paper B the focus was on the integration of KBE with its PLM environment, with a strong focus on IT-related issues regarding interfaces. The focus of Paper C is instead on the effects of an explicit regard for the constraints of the PLM architecture already in the early phases of the development of a new KBE application, which is in line with Research Question 1. From the industrial point of view, however, the aim of this study was to enable the reuse of manufacturing experience in the design phase using a KBE application to enable better support for design for manufacturing.

The industrial needs for the KBE application were identified in a preceding study (Andersson et al., 2008) where it was found that both designers and manufacturing engineers experienced that feedback of manufacturing experience was not efficient enough. The manufacturability of the design was considered to be increased by making it easier to find and access relevant manufacturing experience which, for manufacturing quality purposes, already is stored in manufacturing databases. There is a possibility to reach this information, but the number of different databases is large and the data are formatted and organized to fit the manufacturing context, making the data hard to navigate and retrieve for a designer.

The ambition to incorporate an explicit regard for the constraints of the PLM architecture early in the development process led to an analysis of the process and IT environment (see Figure 4-2). related to the purpose of identifying relevant manufacturing experience and including it in the product design process. A sequence of tasks related to accessing and compiling data from four different databases (all from the manufacturing process) were identified. Also an analysis of IT tools used by the designers at their desktops was made, to ensure that the KBE application did not duplicate any information or any task already carried out somewhere else. In addition, two important architectural principles which were explicitly included are to duplicate no data that reside somewhere else (in this case all of the product-related manufacturing data) and to keep the amount of interfaces to a minimum (Zachman, 1987). The first principle was included with the assumption that manufacturing data was accessed in real time. The second principle was included with the assumption that an SOA was going to be used to integrate the KBE application with the manufacturing databases (inspired by the experiences from Paper B).

The result is a demonstrator which is implemented as a web application (a web page) in which the manufacturing experience is presented in relation to product components and functions, as requested by the interviewed designers. The knowledge encoded into the web application can be primarily categorized as process knowledge related to the process of retrieving the data and information in the right order and from the right sources. The product knowledge is that which is codified in the sources depicted in the bottom, and could be kept outside the code of the KBE application. The ambition to use an SOA for the integration was set back by the unwillingness of corporate IT system owners to provide data from their systems in a neutral information format. For this reason the demonstrator was never integrated with the databases depicted in Figure 4-2.

The conclusion from Paper C is that an explicit inclusion of the PLM architecture affects the development of KBE applications primarily by reducing the need for software development and coding of redundant data and interfaces. A direct effect is that the investment needed for the development of the KBE application can be kept as low as possible since only code necessary for

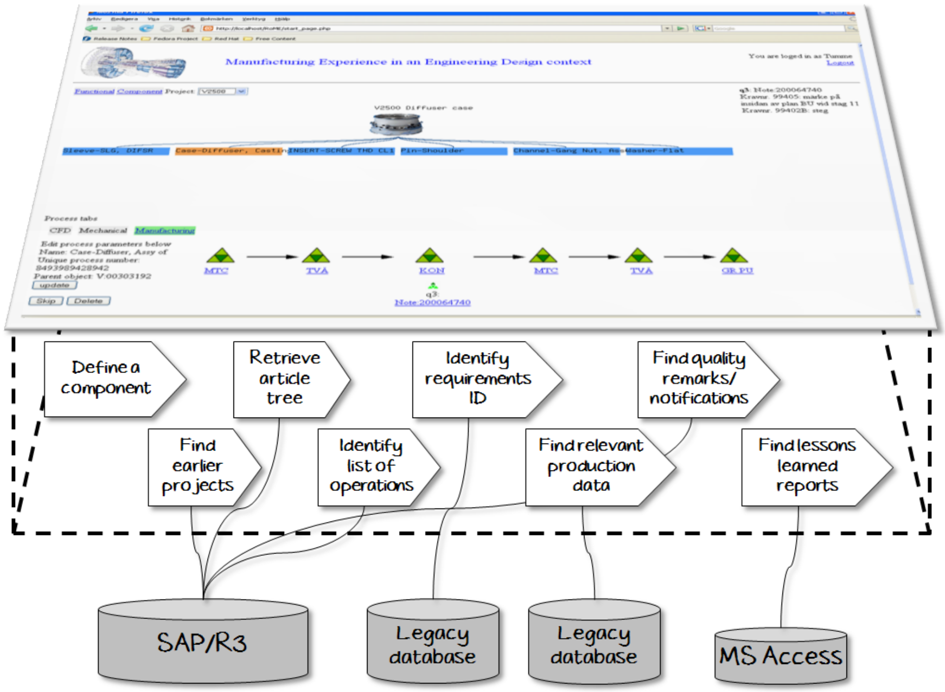


Figure 4-2: Architecture and screenshot of the demonstrator in Paper C

the application workflow and interfaces needs to be developed. A lower number of code lines also implies a smaller need for maintenance and costs involved with it. In addition, easier maintenance implies that when process requirements change the developed KBE application can be adapted with less resources. The experiences from Paper C, however, also show that even though an SOA contributes to tearing down technological barriers related to process and IT integration, these barriers can still be maintained if IT system owners are given relatively strong influence and mandate in relation to business process owners.

4.4 Paper D: Requirements management when introducing new mechatronic subsystems – managing the knowledge gaps

The contributions in Papers A-C reflect the process and IT perspective of KBE and deal with integration and utilization of KBE in product development from that perspective. The recognition of KBE as also a knowledge management method motivated a study of product development from a knowledge management perspective.

The aim of Paper D was therefore to study a product development process in an explorative fashion to investigate the need for knowledge reuse by identifying (1) knowledge that is considered as important and (2) gaps in that knowledge to address Research Question 3. The choice of case for this study fell on a recently finalized product development project at Global Corp. Driveline division dealing with the development of a new commercial vehicle driveline containing a large, and sourced, mechatronic subsystem. The architecture and the main characteristics of the mechatronic subsystem were previously developed in the R&I division of Global Corp. and the technology transfer entailed a great deal of knowledge transfer.

Simultaneously, technology transfer also implies the existence of a knowledge gap in the receiving unit.

The interview results show that, in the particular processes of the studied organization, knowledge about the mechatronic subsystem's functional structure and interfaces was lacking already in the technology transfer and had large effects on the requirements management and supplier management processes. The lack of this particular knowledge has also accentuated the limitations and shortcomings of the component-oriented organization. The respondents' answers revealed that much of this knowledge was built up during the previous phases of development in the R&I unit, as support for decisions taken regarding the architecture of the mechatronic subsystem, but this knowledge was poorly transferred. Furthermore, a larger portion of the process knowledge related to specification of relevant requirements and verification methods with special focus on geometrical interfaces and functional interactions which result from packing and "system effects" in the driveline had to be developed in the Driveline division. The findings from the interviews and workshops indicated that there is little explicit support for managing this knowledge other than regular meetings (called "interface meetings") which were considered as rather time-consuming and inefficient.

The results from Paper D are a set of recommendations regarding the introduction of new and sourced mechatronic subsystems into an existing product. The recommendations are based on the existence of initial knowledge gaps in the receiving unit regarding the mechatronic subsystem. The knowledge gaps can be closed as part of either the pre-development process or the development process. If closed in a pre-development process (e.g. at an R&I unit), transfer of the acquired knowledge is critical. The recommendations are summarized in Table 1 in Paper D.

4.5 Paper E: Knowledge management in mechatronic product development: effective method for creating engineering checklists

Based on the empirical findings in Paper D, the aim of Paper E was to develop a method to manage knowledge that is considered as important but is currently not managed explicitly. The purpose was to formulate an explicit knowledge management strategy that addresses Research Question 3. For the purpose of managing both the product and process knowledge related to interfaces in the Driveline division (which was identified to be important but currently poorly managed), the method of engineering checklists, as demonstrated in the case of Toyota (Sobek et al., 1999), was found suitable to apply.

Even though the literature describes the usage of engineering checklists and what these checklists look like, with examples from Toyota, no guidance is provided on how to create and implement them. The result from Paper E is a method that supports designers in creating engineering checklists for their respective engine components in order to enable the capture and reuse of the previously mentioned product and process knowledge. The method was implemented in a demonstrator in the form of a macro application developed inside a commercial spreadsheet system (see Figure 4-3 for a screenshot).

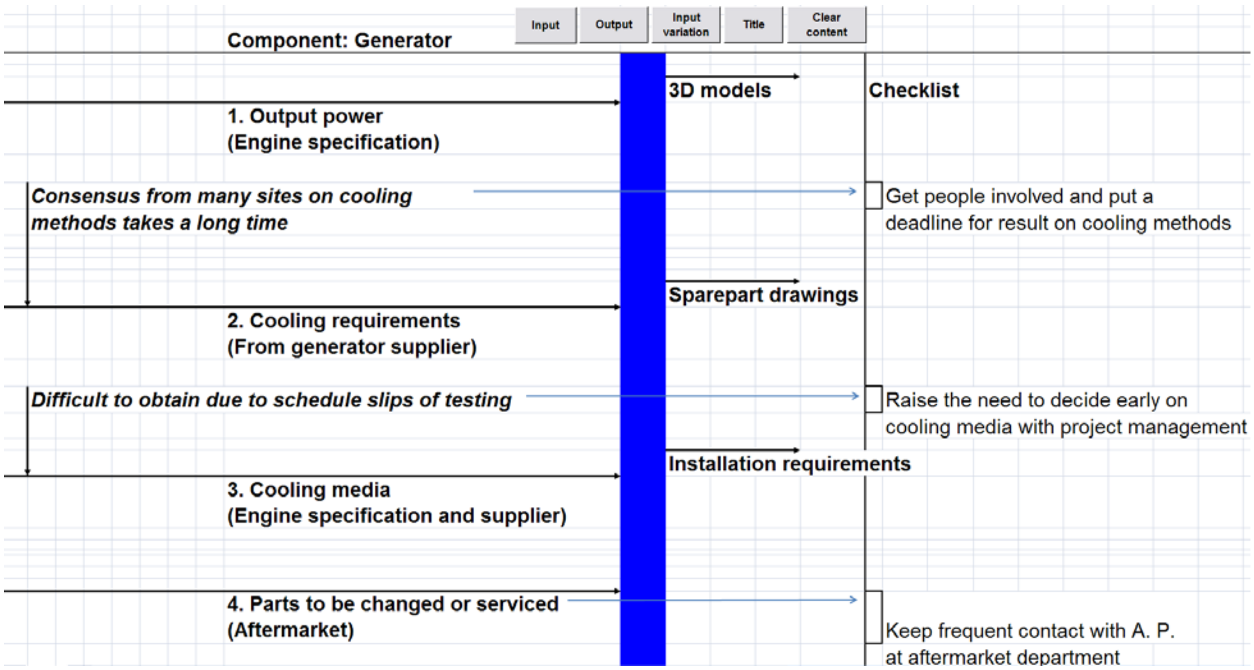


Figure 4-3: Screenshot of the demonstrator in Paper E

The method developed in this paper is supposed to support the implementation of engineering checklists and consists of the following stages and steps:

Stage 1 – Pre-study

The pre-study aims at clarifying the level at which the checklists are to be implemented. The notion of level here refers to how the checklists are delimited and organized. In the studied case a checklist was implemented for each component with a component owner responsible for each checklist.

Stage 2 – Introduction of engineering checklists

This stage involves following three steps whose purpose is to establish the first checklist:

1. Mapping out inputs and outputs related to interfaces and interactions
2. Mapping out causes of lead time related to each of the previously listed
3. Writing a checklist item for each cause of lead time as a reminder of critical or important things to consider.

Stage 3 – Use of engineering checklists

The use of the engineering checklist can be regulated in different ways. It is important that the checklist is used and updated at the same time and that its use is explicitly defined in the process for example in a stage-gate model.

The conclusions from this study are that engineering checklists are a suitable method for managing process knowledge in the detail design phase of a mechatronic product with complex geometrical interfaces and functional interactions. The reasons for this are that they are relatively simple to use and manage and they are compatible with the “stage-gate” thinking deployed in the detail design phase. Though they are simple to use and manage, they still require methodological support for their implementation, which has been addressed by the development of a method that was evaluated with positive results; see Section 6 in Paper E.

4.6 Summary

The results have a clear line of division between Papers A to C – which concern the subject of explicit regard for the constraints of the PLM architecture, enabled by SOA, in KBE – and Papers D and E, which are concerned with the subject of formulating explicit knowledge management strategies in product development. The line of delimitation is explained by the realization that KBE needs to be considered as a knowledge management method to complement KBE development methodologies with this perspective. An overall realization from Papers D and E is that there is a lack of guidance on a methodical approach towards knowledge management in product development. For this reason the lessons learned from the approaches behind the results in Papers D and E, along with an additional study reported in (Ćatić and Malmqvist, 2010b), complemented with aspects from knowledge management literature, have been used to synthesize a framework for formulating explicit knowledge management strategies in product development, which is the main subject of the next chapter.

"A little knowledge that acts is worth infinitely more than much knowledge that is idle."

[Khalil Gibran]

5 Framework for Formulating Explicit Knowledge Management Strategies in Product Development

In this chapter an expedient framework for formulating explicit knowledge management strategies in product development (EFFEKT) is presented. In the last section its relation to KBE is described. The purpose of the framework is to provide a process and guidance on how to formulate an explicit knowledge management strategy in product development. The framework is a generalization of the approaches adopted in Papers D and E and in (Ćatić and Malmqvist, 2010b) integrating lessons learned from these studies with knowledge management models, described in Section 2.2.

The chapter is initiated with the purpose and design of the framework followed by the perspective on knowledge management adopted in order to provide the background and explain certain concepts which are included in the framework. The chapter is continued with an overall description of the different elements of the framework and their purposes. After this the process for the framework is described with focus on tasks, deliverables and guidance. The chapter is rounded off with a discussion regarding how EFFEKT can be used to identify potential for KBE applications in a process or organizational unit.

5.1 Purpose and design of EFFEKT

Before the purpose and the design of EFFEKT are explained in more detail a paragraph is dedicated to discussing the notion of a "knowledge management strategy". Liebowitz (1999) states that a knowledge management strategy can take various forms and exemplifies that a strategy may be to prevent a loss of knowledge due to a 'graying' employee base and thus focuses on a particular part of the organization. The definition of the word "strategy" according to Webster's dictionary is "the art of devising or employing plans or stratagems toward a goal" (Merriam-Webster, 2011). In the context of product development, which is a business process, the "goal" for a knowledge management strategy has to be closely related to the business goals and objectives in order to generate a return on the investment, which is also reflected in prerequisites posed on knowledge management frameworks (Rubenstein-Montano et al., 2001).

Referring back to the example given by Liebowitz (1999) it is clear that his example entails an action but without a business goal for that action. The perspective of this thesis on the notion of a "knowledge management strategy" is that it is the plan of knowledge management actions devised towards the business objectives posed onto a product developing organization. From this perspective EFFEKT constitutes a framework of elements and activities needed to devise and document such a plan.

EFFEKT is based on the premise that product development is knowledge-intensive and that each specific product-developing organization has a unique mix of knowledge types important for that organization's specific processes and deliverables. This mix is further on referred to as a "knowledge application profile" (KAP). EFFEKT addresses the need among product development practitioners to address knowledge management in their processes in a structured and transparent way in order to understand which knowledge is important (and why), to be able to formulate a strategy for managing that knowledge. Based on this need, the general purpose of EFFEKT is to constitute methodological support for practitioners to formulate explicit knowledge management strategies in product development processes. As such, the framework addresses the research gap regarding the general lack of methodological support for addressing knowledge management in product development, described in Section 2.6. The following subsections explain the design of EFFEKT from two different perspectives: as a methodological support in the context of product development and as a framework in the field of knowledge management.

5.1.1 EFFEKT from a methodological perspective

Being a methodological support for product-developing practitioners, EFFEKT has to comply with general guidelines imposed on such or similar support. Norell (1992) has proposed a set of guidelines regarding methodological support for product development which ensure that the support contributes to a more efficient product development process and to a rewarding and stimulating work situation of the individuals. According to Norell (1992) the methodological support should:

1. Be easy to learn, understand and apply.
2. Contain accepted, non-trivial knowledge within the area of current interest.
3. Provide support in order to identify weak spots.
4. Be rewarding to use for different disciplines, leading to the establishment of common references and shared views.
5. Support co-operation and facilitate a learning effect for users.
6. Contribute to a systematic work procedure.
7. Have a positive and preferably measurable effect on the outcome of the product development work within the area of current interest.

Though these guidelines are geared towards methodological support focused on the product development process, they are also applicable to methodological support focused on knowledge management in the context of product development processes.

The elements of EFFEKT and their interrelations are depicted in Figure 5-1. The framework elements are divided into three main parts indicated with numbers: knowledge application (1), knowledge capture (2) and knowledge development (3). Each of these has two elements: process integration and methods and IT, making up a total of six elements in the framework. The idea behind this design is to clearly separate knowledge application, capture and development in order to separately discuss the way each of those is (or should be) integrated with the business process and the way each is supported with methods and IT. The purpose of separating these discussions is to be able to focus on each part without being disturbed by exactly how the elements from some other part are to be realized. For example, if a discussion regarding how knowledge about component requirements is applied is affected by a simultaneous discussion of how that knowledge can or should be captured, then focus is taken away from grasping the role of this type of knowledge in the process and the way it is preferred to be applied. Two additional purposes exist for this separation. One is related to Norell's (1992) third bullet. By separating the three parts one can more clearly identify weak spots in how knowledge is applied, captured or developed in a company. The other purpose is related to Norell's (1992) fourth bullet. By separating the three parts it is easier to establish a common reference and a shared view among the individuals involved in the knowledge management strategy, regarding e.g. why certain types of knowledge are captured (and how) based on why they are applied (and how).

In order to address Norell's (1992) bullets three, four, five and six EFFEKT's design is complemented with a process illustrated in detail in Figure 5-5. The relation between the EFFEKT elements and the EFFEKT process are illustrated in Figure 5-2. The purpose of this process is to guide and support the practitioners through the discussions and establishment of the

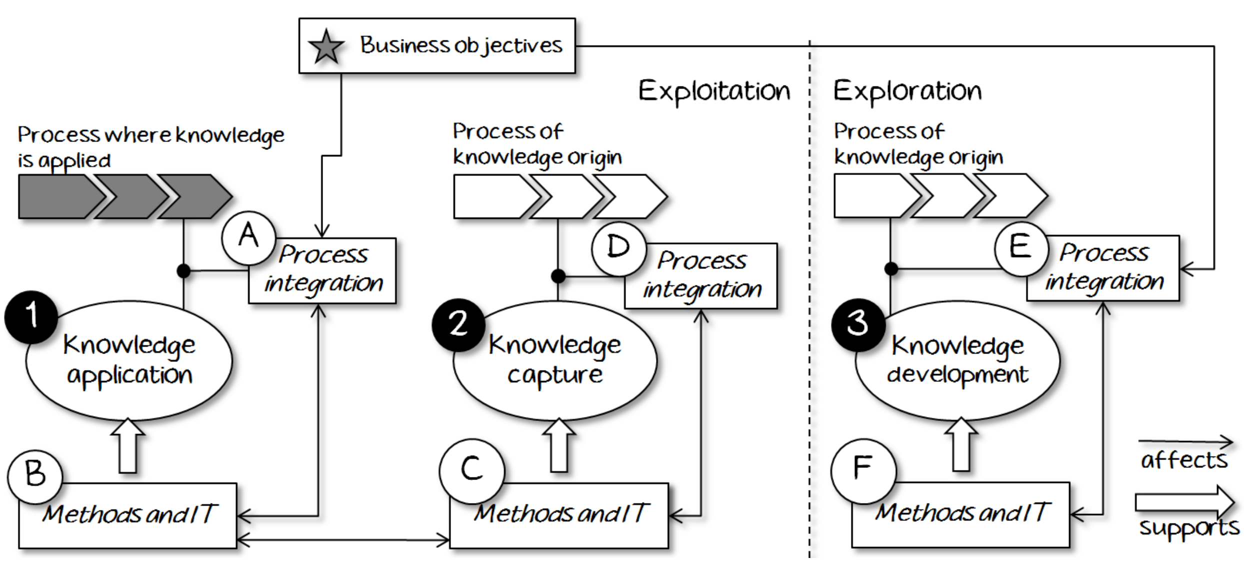


Figure 5-1: The elements of EFFEKT and their interrelations

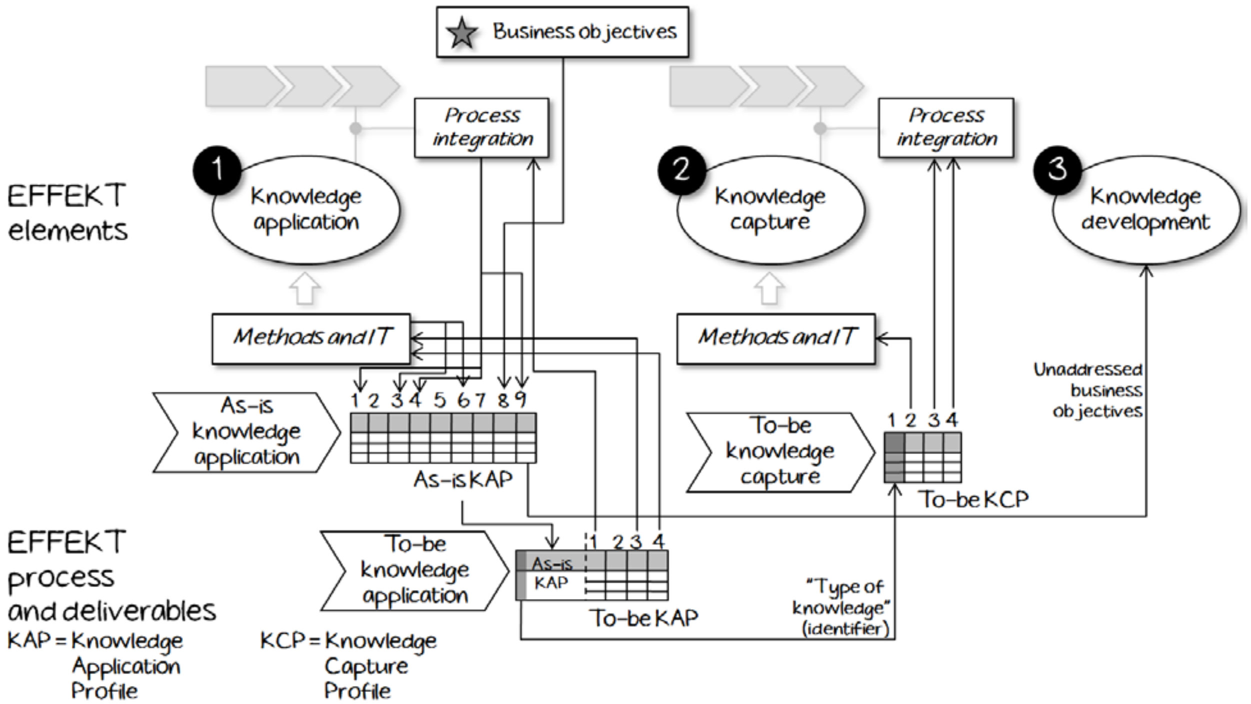


Figure 5-2: The relation between EFFEKT elements, process and deliverables

EFFEKT elements in the context of a specific product development process or organizational unit by providing a sequence for how the different elements are determined and how the discussions are documented. The first elements to be established are the process integration and methods and IT support of current knowledge application (i.e. "as-is"). The reason for starting with this is to obtain a clear view of which types of knowledge are important for the product development process in question. After this, the "to-be" of knowledge application is discussed and agreed upon. Both of these discussions are guided by, and documented in, the as-is and to-be KAP, both illustrated in Figure 5-6 and Figure 5-7, respectively. By doing these two tasks, both the product development process participants and the manager/researcher/expert/consultant who is responsible for managing the EFFEKT process can obtain a clear and documented view of which knowledge is most important to manage, the way that knowledge is currently managed, issues surrounding its management, and how it is desired to be managed instead, all from a knowledge application perspective. An accurate view of how knowledge is applied can also guide the establishment of an accurate view on where and how that knowledge is or should be captured. Since the applied knowledge can originate in other business processes and organizational units, this is also identified. It should be noted that Figure 5-1 also contains elements related to knowledge exploration which are included and discussed in EFFEKT but not explicitly supported with any activities or deliverables in the EFFEKT process.

Besides documenting the current situation of knowledge application and capture, and contributing to Norell's (1992) third and sixth bullets, the different knowledge profiles also contribute to Norell's (1992) fourth bullet mainly by leading to the establishment of a common view regarding the topics of knowledge application and capture. Furthermore, the documented profiles are necessary for the development and implementation of the methodological and IT support for knowledge application and capture.

5.1.2 EFFEKT from a knowledge management framework perspective

In addition to Norell's (1992) guidelines, EFFEKT needs to comply with guidelines related to it as a framework in the context of knowledge management. Rubenstein-Montano et al. (2001) have reviewed 25 different knowledge management frameworks and set forward recommendations for a unifying knowledge management framework. The conclusion from Rubenstein-Montano et al. (2001) is that a knowledge management framework should:

1. Be both prescriptive and descriptive. Prescriptive means that it should state elements such as knowledge management activities that need to be included. Descriptive means that the design and choice of elements need to take into consideration characteristics of the organization such as business objectives and processes, roles and culture etc.
2. Knowledge management activities must be consistent with systems thinking through:
    a) Linking knowledge management to business strategies and objectives
    b) Planning before knowledge management activities are undertaken
    c) Aligning knowledge management with the organizational culture
    d) Including iterative processes directed by feedback and learning
3. Include activities for finding, verifying, storing, organizing, sharing and using knowledge
4. Cater for a distinction between tacit and explicit knowledge making sure each is handled appropriately
5. Accommodate both single-loop and double-loop learning

Since EFFEKT is a framework for formulating knowledge management strategies, it aids in the establishment of particular knowledge management frameworks. With this in mind it is important that the output from EFFEKT complies with the above-stated recommendations. In regard to the first recommendation, EFFEKT clearly highlights the need for a knowledge management strategy to reflect the needs and characteristics of the organization, which implies that the strategy is descriptive. The output of EFFEKT entails also prescriptive elements, embodied in the to-be knowledge application profiles. The purpose of EFFEKT is to make sure that the knowledge management strategy is formulated in its context, which is in line with the second recommendation (especially with 2b). The recommendation 2a is explicitly included while 2c and 2d are implicit. The aspect of culture is not explicitly mentioned, but is reflected in the different knowledge profiles stating how knowledge is or should be applied and captured. The recommendations 2d and 5 are addressed implicitly through the elements concerned with knowledge development. Learning as a phenomenon, however, has not been explicitly addressed because product development is considered to be a knowledge-intensive process, implying that learning is taking place as a part of engineering activities. The activities in the third recommendation are also implicitly included in the groups labeled "knowledge application", "knowledge capture" and "knowledge development" but are not elaborated in detail. Finally, the fourth recommendation is addressed mainly through the inclusion of personalization and codification in the method column of the knowledge profiles, where a personalization approach indicates management of tacit knowledge and codification indicates management of explicit knowledge.

5.2 Adopted perspective on knowledge management in EFFEKT

Adopting the perspective of an individual engineer or a team of engineers, there are two major ways in which knowledge explicitly can be applied and help the product development process to progress: reuse and consultation, as illustrated in Figure 5-3. These two ways reflect two different viewpoints on knowledge in product development: knowledge that is created in the context of a certain process or product, and knowledge that is created in another context but can be applied to support the context of a certain process or product. In extreme cases, the difference in contexts can reflect different kinds of products in different companies, but a more common situation, as observed in the appended papers, is that the difference in context reflects different functions, such as Driveline division consulting the R&I division (Paper D) or design engineers consulting manufacturing engineers (Paper C).

This perspective means that it is the application of the knowledge that is focused upon and constitutes the starting point for the formulation of the explicit knowledge management strategy. The reason for adopting this viewpoint is to align the knowledge management strategy with the business process in order to make sure that the knowledge to be managed is present in and has an effect on the business process. An important aspect that is, however, left out of scope for this thesis is how the context of knowledge application is delimited. In an extreme case one might end up with a knowledge management strategy for every individual engineer, and at the other extreme is a knowledge management strategy that defines the complete company as one context in which a large variety of knowledge types have to be managed. Based on the experiences from Papers D, E and from (Ćatić and Malmqvist, 2010b), an appropriate delimitation of context is a

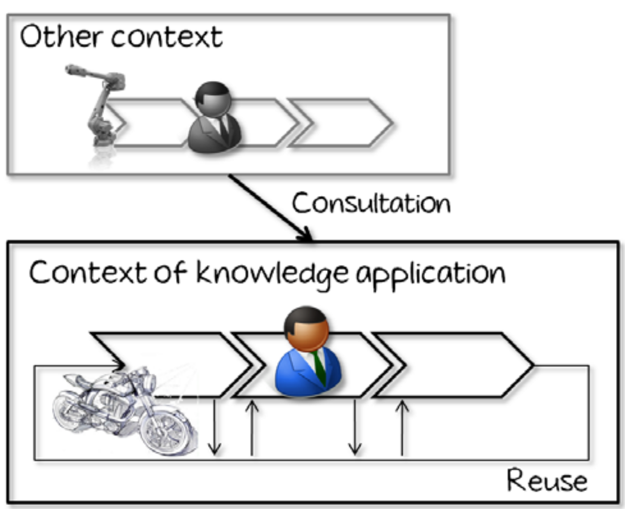


Figure 5-3: Knowledge application through reuse and through consultation

group of engineers working either in the same process (e.g. project managers, calculation engineers, test engineers), with the same product subsystem (e.g. base engine or control system) or with the same product functions (e.g. crash safety or vehicle dynamics).

From the perspective of an individual engineer or team of engineers working in a defined context, there are several types of knowledge that can be applied. The categorization of the different types of knowledge that constitutes the foundation for the proposed framework is depicted in Figure 5-4. This categorization is based on that of know-what, -how, -why and -who combined with the division into product and process knowledge, as illustrated in Figure 2-8. One more categorization is superimposed onto these: the division between solution and information, which adds up to a total of 16 knowledge categories. In Section 2.3 regarding product development processes, it is stated that product development processes output both a product solution and information describing that product solution to a certain level of detail so that the product can be produced, legislated, marketed, sold and serviced. There are two levels at which knowledge application can take place, when reusing the product solution and when reusing the product information, both of which have been observed in the studies behind Papers B, C, D, E and (Ćatić and Malmqvist, 2010b). An example observed in Paper D is that the product solution could evidently not be reused (because a large portion of the new driveline was completely new). But requirements specifications of similar components (product information) were reused and the engineers found them hard to reuse because the knowledge behind those documents was lacking. The same kind of reasoning can be applied to processes. An example from a workshop held regarding the development of the wiki in (Ćatić and Malmqvist, 2010b) is that, in many projects, different kinds of test methods (process information) with corresponding test equipment (process solution) are developed. Sometimes there is a need to reuse the information to develop new equipment, and sometimes there is a need to reuse existing equipment.

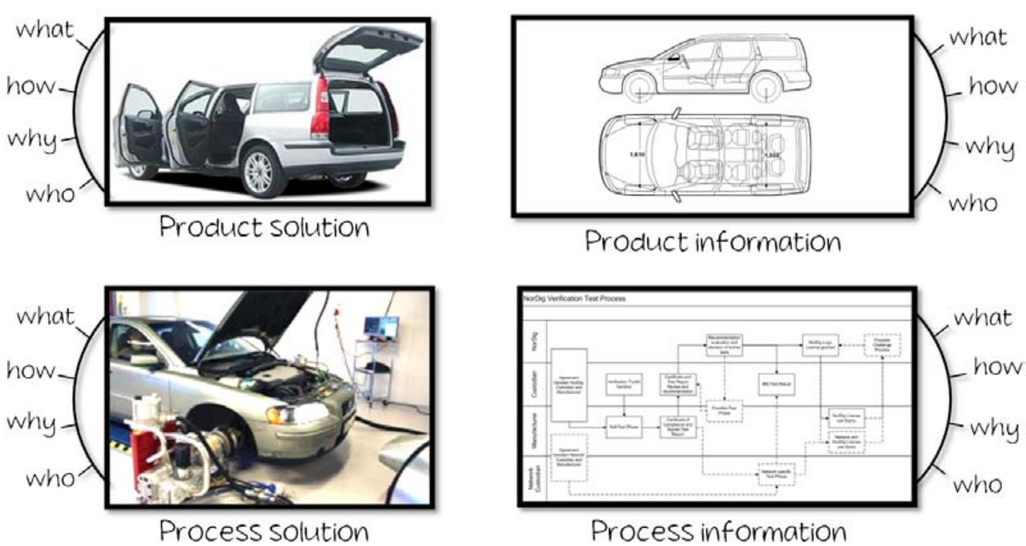


Figure 5-4: Knowledge categorization used in EFFEKT

In addition to these categories, the dimension labeled “level of formalization” in Figure 2-8 is included in the framework. The reason for this is that the desired level formalization has a direct effect on the methods and tools that can be implemented to support the knowledge management strategy. In addition, if the desired level of formalization significantly differs from the current level of formalization in the management of a certain type of knowledge, then this also has an effect on the effort and resources needed to support the desired knowledge management strategy.

5.3 EFFEKT elements

The elements of the framework are divided between two areas reflecting knowledge exploitation and knowledge exploration. These two areas are divided into three different parts: knowledge application (1), knowledge capture (2) and knowledge development (3). Each of these has the elements “process integration” and “methods and IT”.

5.3.1 Exploitation

When establishing the exploitation part of the knowledge management strategy, there is a need to harmonize the elements related to knowledge application and knowledge capture with each other. In order to do so, the elements of process integration with supporting methods and IT have to be carefully and explicitly designed to ensure that there are no inconsistencies between them in terms of which knowledge is captured and applied, how it is captured and applied, and using which methods and tools. For example in Paper C the knowledge regarding the manufacturability of the product design was captured in a format that suited the manufacturing process but was hard to understand for design engineers. The purpose of the developed demonstrator was to harmonize between the elements of *“C”* and *“B”* by compiling and presenting the knowledge in a format which is logical from the perspective of knowledge application. The elements of *”D”* and *“C”* already existed, because there were computer terminals with installed software whose usage was regulated by the quality control process in manufacturing, in order to document every incident which had a negative effect on either the product quality or the manufacturing operations. Also the process integration of the knowledge application (element *“A”*) was explicitly formulated but poorly supported by element *“B”*. Therefore it can be argued that an explicit strategy to manage this knowledge was in place but the elements were not in harmony.

This thesis argues that one should start with knowledge application in order to understand which types of knowledge are of higher significance from the perspective of the business process. By focusing on knowledge that benefits the process where it is applied, a “pull” for knowledge is obtained and it is easier to argument for why this knowledge should be captured. The explicit formulation of which knowledge is or should be applied and how, to meet the business objectives, makes up the element *“A”* in Figure 5-1. To support the discussion and documentation (in a knowledge application profile) regarding the types of knowledge that are or need to be applied, the categorization illustrated in Figure 5-4 can be used. When it is decided which types of knowledge are to be applied and how, element *“B”* – methods and IT through which knowledge application is supported – can be determined and documented in the knowledge application profile.

After this is completed, one has to examine where the different types of knowledge applied originate and how they are or should be captured. The two different ways of applying knowledge, illustrated in Figure 5-3, imply that "reuse" requires knowledge to be captured internally in the same process where it is applied, while "consultation" requires the knowledge to be captured in a different process than where it is applied. Regardless of which, the framework highlights that the methods and IT used for the knowledge capture have to be in harmony with those used to support knowledge application. An issue of discord can be caused by e.g. the format in which knowledge is captured, which was the case in Paper C. Discord can however also be caused by more fundamental aspects, for instance if a codification approach is desired in knowledge application while a personalization approach is desired for knowledge capture. If the ways in which knowledge is desired to be applied and captured are not in harmony, there is a risk of motivational barriers appearing (Dixon, 2000), as described in Section 2.2.3.

The element of methods and IT for knowledge capture is closely related to which knowledge needs to be captured and how its capture is integrated with the process, i.e. element *"D"*. Taking the example from Paper C again, one of the prerequisites for knowledge capture in the manufacturing process is that whenever an incident occurs the process requires the operator to document this. In Paper E this issue was discussed in terms of how often the engineering checklist needs to be updated and whether it should be connected to the project process or component release process. In (Ćatić and Malmqvist, 2010b) the aspect of how the time used for knowledge capture was to be financed came up, which is another significant aspect of the process integration.

An example of the whole process (from *"A"* to *"D"*) can be found in Papers D and E when combined. Paper D started with a mapping of which knowledge is applied and how, using which methods and IT along with an analysis of which types of knowledge need to be managed differently and how. It was concluded with a set of recommendations related to how to improve knowledge application both through consultation (mainly in relation to the R&I division) and through reuse. Paper E continued with the development and implementation of a method to support reuse of knowledge related to component interfaces, which was found to be important for the product development process but currently only implicitly managed through so-called "interface meetings" which were considered to require too much time.

5.3.2 Exploration

The elements described so far reflect the left half of Figure 5-1, which has to do with exploitation of existing knowledge or knowledge that is created simultaneously with the execution of the product development process. Just as noted by March (1991), an organization has to engage in knowledge exploration to some degree to secure the long-term survival of the company. For this reason EFFEKT includes such elements as well. In the same way as knowledge application and knowledge capture need an explicit process integration supported with methods and IT, the same is necessary for knowledge development. Similarly to knowledge capture, knowledge development can be done within the same context in which the knowledge is applied, or it can be done in some other context.

The most likely trigger for knowledge development is the realization that the current level of knowledge is not sufficient to meet the business objectives. An example is the case from Paper D where the objective to significantly lower the fuel consumption resulted in the need to develop new product solutions and consequently to also develop new product and process knowledge related to those solutions. In the case of Paper D the product knowledge was developed in an external context (that of the R&I division) while the process knowledge was developed simultaneously as the studied product development project was executed (which was one of the reasons for why the project faced issues with schedule slips). The element of process integration (element *"E"*) in exploration means that resources and time are dedicated to activities whose primary purpose is to elevate the level of knowledge in relevant areas. The way knowledge development is integrated with the business process affects and is affected by element *"F"* which comprises the methods and IT applied to support the knowledge development.

Within the Lean Product Development framework, it is advocated that knowledge development takes place in the same context as knowledge application but clearly separated in time, which is one way to approach element *"E"*. The notion of "knowledge gaps" is used as an indicator for when dedicated knowledge development activities need to take place. Usually this happens when it is realized that a decision-related to the product design cannot be taken due to high uncertainty in the knowledge and information used for that decision. When this happens the process requires that activities dedicated to knowledge development take place (e.g. simulation, building of simple prototypes or similar depending on the decision and the uncertainty) in order to ensure that the decision is based on valid knowledge and not assumptions. The Lean Product Development framework also has a more explicit view on which methods should be used to support knowledge development (primarily testing of prototypes) and also to document it (primarily using different types of A3 reports and trade-off diagrams). The use of these methods is explicitly governed by the process.

5.4 EFFEKT process

The EFFEKT process is illustrated in Figure 5-5. Its main tasks and deliverables are described below, and then each task is elaborated in the following sub-sections in more detail and with respect to the framework elements. The process should be managed by either an external consultant/researcher or an internal dedicated resource, and it should be supported by the manager of the aforementioned organizational unit. It should be noted that the proposed framework process is not executed once but is carried out continuously, to make sure that the knowledge management strategy reflects the needs of the business process.

1. Determine the "as-is" state of knowledge application

The aim of this task is to determine the way knowledge is applied in the process and document it in the "as-is knowledge application profile" whose details are described in Section 5.4.1. The purpose of this task is to map the different types of knowledge that are applied, their roles in the process, their origin, how they are applied, and using which methods and IT. For example in the case study from Paper D most of the knowledge applied was regarding requirements and the requirements management process. Consultation with engineers from other departments through telephone calls was the dominating method, in which the main issues were related to identifying the right individual who could answer a question. The inputs for this task are documentation of process maps and product development process deliverables. The methods applied are interviews and workshops together with the unit members, as well as analysis of deliverables and observation on behalf of the consultant or researcher.

2. Determine the "to-be" state of knowledge application

The results from the "as-is" analysis documented in the knowledge application profile are used as input for this task along with the business objectives for the organizational unit or process in question. The purpose of this task is to compare the knowledge types identified and their relation to the business process and the business objectives to determine the priority of managing each particular type of knowledge. The "to-be knowledge application profile" also seeks to explicitly define how the different knowledge types are to be managed in terms of methods and IT.

In addition, another purpose of this task is to identify business objectives which are not addressed with existing knowledge types and initiate the development of those knowledge types. This kind of task was carried out between Papers D and E, where the recommendation to the studied department in the Driveline division was to focus on developing process knowledge related to their subsystem. Also the decision was taken to focus on a more explicit management of interface related knowledge supported with a dedicated method (described in Paper E).

3. Determine the "to-be" state of knowledge capture

The input for this task is the "to-be knowledge application profile" and the purpose of this task is to explicitly define how the capture of the knowledge that is desired for application is to be integrated with the business process from which the knowledge originates. Another purpose is to define which methods and IT to use for knowledge capture based on the way knowledge is desired to be applied.

5.4.1 As-is knowledge application profile documentation

As mentioned in the introduction of this chapter the knowledge application profile is the unique mix of knowledge types which are important in a certain process or organizational unit. The profile is documented using the form illustrated in Figure 5-6. The form consists of a set of

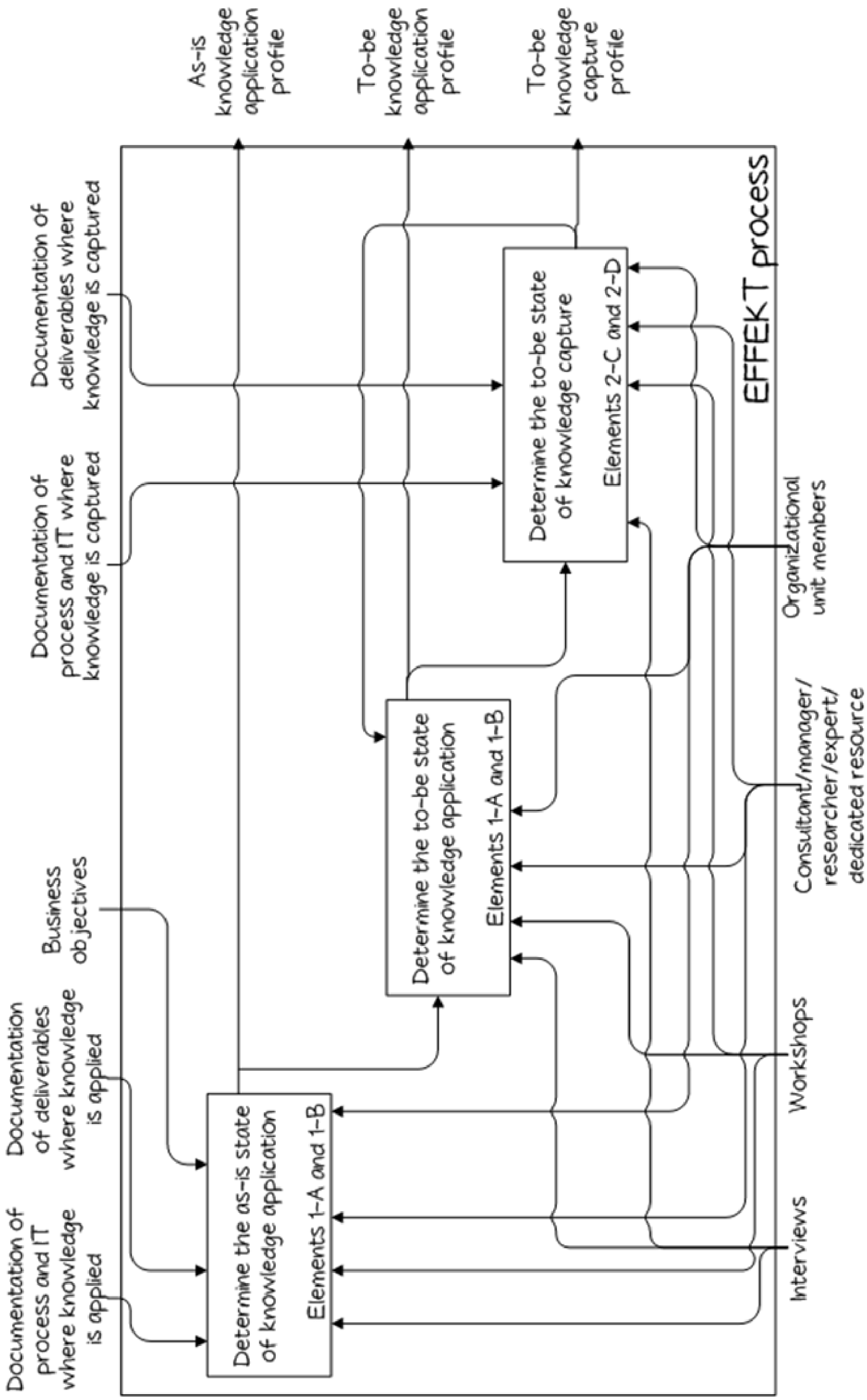


Figure 5-5: IDEF0 diagram of the EFFEKT process. Process-IDs refer to Figure 5-1.

columns which are related to the framework elements in the way illustrated in Figure 5-2. The purpose of the as-is knowledge application profile is to document the current state of knowledge application either through interviews or workshops (or both) together with the organizational unit members in order to achieve an agreement in which types of knowledge that are important for the process, how they are applied, how they relate to the business objectives and which problems (if any) the organization members experience with their application. Each of the columns is described in more detail below.

Task/deliverable

The documentation regarding the deliverables and tasks of the organizational unit are used. The purpose is to make it easier to reason about the types of knowledge applied as they are put in relation to either the tasks or the deliverables (or both). The purpose is not to produce an exact and complete list of tasks and deliverables, because the primary aim is not process mapping but the gaining of an understanding of the knowledge involved in the process.

Type of knowledge

Using interviews and workshops the organization unit members can be asked to reflect upon which types of knowledge they apply in relation to the listed tasks and deliverables. This kind of inquiry can reveal those types of knowledge which are explicit and can be stated by the members with more or less stimulation. The types of knowledge that are implicit or tacit and cannot be expressed by the members have to be inferred by analyzing the tasks/deliverables and comparing them with the stated knowledge types to identify possible other types of knowledge that are involved. These in turn can be used to stimulate the extraction of further types. This is looped around until all the listed tasks and deliverables have at least one type of knowledge related to them. It is important that no restriction is given on what a “type of knowledge” is. The members should be given the freedom to answer however they feel. Existing knowledge categorizations can be used as stimulation but not if they are found restricting.

IT support

The IT support is that which is involved in the performance of the task or related to the deliverable. The purpose of listing this is to establish a view of the IT support involved in the process, in order to be able to reason about which existing IT tools or systems that knowledge

Task/ deliverable	Type of knowledge	IT support	Reuse/ consultation	Knowledge category	Method	Level of formalisation	Business objective	Problems

Figure 5-6: As-is knowledge application profile

application potentially could be integrated in. If dedicated knowledge management tools (such as e.g. KBE) are to be developed then this list helps to identify with which IT tools or systems the new tool needs to integrate, interface or interact with.

Reuse/consultation

For each identified knowledge type it should be indicated whether it originates from the context of the process execution or if an external part is consulted. The external part can be either another person but it can also be a database (as was the case in Paper C).

Knowledge category

The aim of the knowledge categorization is to harmonize among the previously listed types of knowledge. The purpose here is that the categorization can guide the selection of suitable methods and IT to support the knowledge management strategy. In this thesis the categorization illustrated in Figure 5-2 is used, but the framework is in no way limited to this categorization if some other categorization should be identified as more suitable or easy to use for the guidance of appropriate methods and IT solutions.

Method and Level of formalization

For each of the identified knowledge types the method currently employed to support the application should be stated. The method used will also reveal whether it is a question of personalization or codification approach; e.g. e-mail or telephone call indicates personalization, while lessons-learned reports or design guidelines indicate a codification approach. The level of formalization should be stated according to a predefined scale such as “1 to 5” or “low, medium, high” in which it is indicated what the different levels mean, e.g. low means that a person provides the answer, medium means that a report is written and high implies the use of rules and formulas.

Business objective

The input of business objective referred to here is a concretization of the business objectives of the overall organization in the context of the focused organizational unit. Usually the overall strategic objectives are stated at a high level of abstraction such as “lower cost”, “shorter lead times”, “higher quality”, “increased profit” and so on. Depending on the specifics of the focused organizational unit in terms of processes and deliverables, the overall objectives have to be formulated more concretely. As an example we can take the case from Paper C where lower product cost and higher product quality are directly related to how well knowledge regarding the manufacturing process is applied in product development. The purpose of including this aspect is to motivate and create a consensus among those involved in the knowledge management strategy as to why certain types of knowledge need to be explicitly managed.

Problems

The problems involved with the application of each knowledge type should be listed here. It might be that a person is hard to access, a document is poorly or differently formatted, or that a guideline is hard to interpret. Any problem with the application of an identified knowledge type should be stated.

5.4.2 To-be knowledge application profile

Once the current state of knowledge application is agreed upon and documented it is time to discuss how the identified types of knowledge can be applied in a better way in order to improve their management or in order to improve the way they support the business process. The column where the problems are stated can be one way to start the discussion. Another way can be to start with the business objectives to focus the discussion regarding knowledge types which are related to objectives which are of higher importance. Regardless of approach, the purpose of is to discuss whether the unit members wish to change the way certain types of knowledge are applied and, if so, how the desired way of application is. This discussion is documented in the to-be knowledge application profile which is based on the as-is KAP with an addition of four columns, as illustrated in Figure 5-7.

Priority

In order to determine how the knowledge is to be applied the first task is to make a prioritization among the knowledge types from the as-is KAP. The purpose of this is to indicate which types of knowledge that are considered to be more important and whose management should be assured first.

Business gain

In the case that the organization unit members agree on changing the way one or several types of knowledge are applied the main gain from the business perspective should be documented in this column in order to provide an argument for why resources should be invested.

Difference in Level of formalization or in Method/IT

A change in the way knowledge is currently applied mainly has effects on the level of formalization and/or in the method and IT used. This effect is documented as the difference in either of these aspects that a change of knowledge application implies. An example from Paper C is that the personalization approach which involved the inclusion of manufacturing engineers in design projects to assure application of manufacturing knowledge was considered poor. The to-be strategy was to shift to codification since much of their knowledge was already codified.

In this step it is also important to consider how the complexity of different IT tools and systems can be minimized. If for example an existing tool (e.g. a PDM system) is already managing

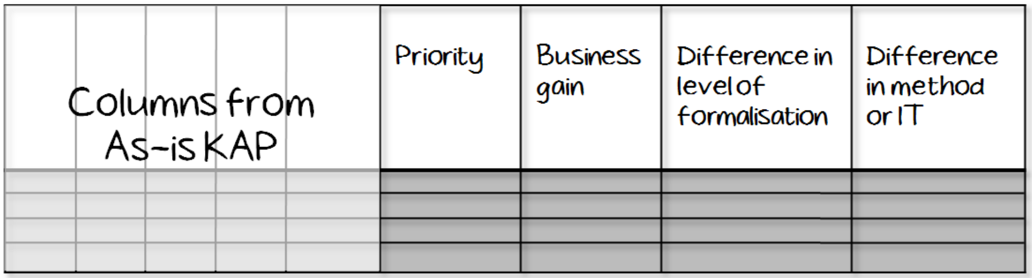


Figure 5-7: To-be knowledge application profile

identified types of product know-what in a satisfactory way, other types of product know-what can be considered to be integrated there. If this, however, is a poor solution from the perspective of knowledge capture then this should be seen as a constraint (as illustrated in Figure 5-5). Some looping between the method and IT in the "to-be knowledge application profile" and the "to-be knowledge capture profile" is necessary, also indicated by the double-headed arrow in Figure 5-1 between the elements 1-B and 2-C.

5.4.3 *To-be knowledge capture profile*

When the details of current and desired knowledge application are documented it is time to focus on how the different types of knowledge applied are captured. The purpose of the to-be knowledge capture profile is to document which types of knowledge that are captured and how as well as how the capture of each knowledge type is integrated with the process of knowledge origin. For the types of knowledge which are reused (as indicated by the fourth column in the as-is KAP) the organizational unit members can discuss the knowledge capture since they are involved in it while for knowledge types that are applied through consultation the issue of knowledge capture has to be discussed with the members of the process/organizational unit where the knowledge is captured. In order to ensure the traceability between knowledge capture and application the knowledge type column is kept as identifier. The details of the "to-be knowledge capture profile" are illustrated in Figure 5-8 and more closely described below.

Method and IT

The approach adopted for knowledge capture, supported with methods and IT, has to be in harmony with the approach supported with methods and IT in knowledge application. This is necessary in order for the captured knowledge to be applicable; otherwise one risks capturing knowledge that is of limited value when applied. In reality, however, this reflects an idealistic situation. The details in the "to-be knowledge application profile" have to be weighed against the following aspects:

Technical aspects

Technical aspects relate to issues such as access to necessary equipment, information systems, education etc. For example: do those who will act in the knowledge capture have access to

Type of knowledge (identifier, from KAP)	Method & IT for capture	Source	Related to task/ deliverable in the process of knowledge origin

Figure 5-8: To-be knowledge capture profile

appropriate tools to be able to "feed" the method and IT which are planned to support the knowledge application? In the example from Paper C, manufacturing experience already exists because each machine operator on the production floor has a computer terminal and is required to report incidents and quality remarks. If they instead wrote in a notebook, the manufacturing experience would not have been accessible using any digital tool. Issues related to the IT architecture also fall into this category, where any chosen IT solution has to comply with the surrounding IT environment.

Practical aspects

Practical aspects concern time allocation, distances, time zones, personal preferences and so on. For example, if a personalization approach is desired then one should make sure that there are enough people with enough dedicated time for it to work. The example in Paper D regarding the "interface meetings" shows that there is not enough time for a personalization strategy to be feasible.

Economic aspects

The economic aspects make sure that the cost of knowledge capture does not outweigh the gain of knowledge application. If too much time or money needs to be invested in activities or IT tools for knowledge capture then there is no financial soundness in investing.

In order to address these aspects and at the same time satisfy the need for harmony between knowledge capture and knowledge application it is important that an iterative approach is adopted, which is indicated in Figure 5-1.

Source and Task/deliverable

Once an acceptable solution is identified and an approach supported with methods and IT is in place for the capture of each type of knowledge, it is important to explicitly relate it to its source and task/deliverable in the process where the knowledge originates. The aim of this is to establish a proper integration of the knowledge capture with its process of origin. The purpose is to identify which type of knowledge has to be specified as an explicit deliverable of which task or as part of another deliverable. The risk is otherwise that the knowledge capture is not planned for with dedicated time and resources, and is either forgotten or omitted, thus jeopardizing the sustainability of the knowledge management strategy.

5.4.4 *Knowledge development (EFFEKT element 3-E and 3-F)*

The case where some business objectives do not have any knowledge types which contribute to them or are not addressed extensively enough indicates the need to develop or acquire new knowledge, as illustrated in Figure 5-2. The realization that new knowledge needs to be developed does not necessarily have to surface during the first execution of the framework process. The need to develop new knowledge can also be a result of a change in business objectives. This activity reflects an exploration strategy, and the notions of methods and IT mentioned in this context have very little to do with those in the exploitation context.

Strategy

The strategy for knowledge development should be decided based upon the type of knowledge that needs to be developed and how it relates to the unaddressed business objectives of the organization. A second aspect for consideration is whether the knowledge should be developed internally or externally, as the example from Paper D indicates that the business objective for reaching significantly lower fuel consumption resulted in an external development of product knowledge in various R&I projects, while the process knowledge for integrating the mechatronic subsystem was developed internally due to the roles of the respective division in Global Corp. An extreme form of external knowledge development is the purchasing of technologies or hiring of specialist consultants. A third aspect to consider, and which was noted during the study behind Paper D, is that already in the planning of activities for knowledge development it is important to establish a strategy, supported with methods and IT, for the application and capture of the types of knowledge that are expected.

Methods

Among the methods for knowledge development can be mentioned dedicated research projects, advanced engineering and technology development. Lean Product Development literature advocates a systematic utilization of testing and simulation as methods for knowledge development.

IT

The primary way to use IT for knowledge development is to use digital models and perform simulation to develop new knowledge about the product and the process. Some support for this approach can be found in commercial CAD and CAE systems where a certain property, e.g. stiffness, for a component or assembly is assessed as a function of a certain design parameter, e.g. a thickness or an angle. The system performs a series of simulations and presents the result, thus enabling the designer to understand trade-offs and dependences in the design. The main constraint of simulation-based knowledge development is that knowledge can only be developed within the boundaries of what can be simulated. Moreover, the validity of the knowledge is directly related to the validity of the simulation model.

5.5 Identifying potential for KBE with EFFEKT

The purpose of the presented framework has been to support an approach in which an explicit knowledge management strategy is formulated on the basis of an organizational unit's characteristics in terms of processes, deliverables and business objectives. The different knowledge profiles are used to ensure coherence and traceability between organizational/process needs and methodological/technological enablers. Discussion regarding specific knowledge management solutions has been kept at a rather general level because the solutions depend on the characteristics of each individual case. Since the original starting point for this thesis is to contribute to a wider utilization of KBE in product development, it reflects a "push" approach as seen from the perspective of the framework. The purpose of this section is therefore to elaborate on which attributes in primarily the knowledge application profiles characterize cases with potential for KBE. Relevant columns from the two knowledge application profiles are addressed in the following subsections.

Task/deliverable

In the assessment of tasks or deliverables listed in the as-is knowledge application profile, the graph proposed by Sunnersjö (1994) and illustrated in Figure 1-2 can be used. Sunnersjö (1994) suggests that increased process maturity indicates potential for KBE from a "task" point of view, and that increased product variability indicates potential for KBE from a "deliverable" point of view. As mentioned earlier, both of these are reinforced by Pugh (1996) who argues that potential for KBE rises with the stability and maturity of the underlying product concept, which refers to both the tasks and deliverables. The conclusion is that if the tasks are repeatable and mature and/or the deliverables related to the product design are repeatable and mature, this indicates potential for KBE because the repeatability indicates that there is a financial gain in investing, and the maturity indicates that the knowledge base is not likely to change, keeping the maintenance costs down. Besides this, the listed tasks can also be compared to the task categories suitable for KBE as suggested in the frameworks of KLIC (Guida and Tasso, 1995) and CommonKADS (Schreiber et al., 2000).

In addition to the already existing recommendations, the notion of "value addition" is suggested here as an additional aspect to consider in respect to listed tasks. The Lean product development paradigm defines three types of tasks: value-adding, non-value-adding but necessary and wasteful (Ward, 2007). During the research studies I have encountered the mindset that activities which are perceived as wasteful should be automated in order to be "eliminated" using KBE. What is important to remember is that KBE does not eliminate tasks, and that one will have to make an initial investment and a continuous maintenance which is also associated with costs. If an activity is proven to be wasteful, it should be eliminated by exclusion from the processes and not by automation. For these reasons KBE applications should be applied only to value-adding tasks (such as design or simulation) or non-value-adding but necessary tasks (such as certain kinds of preprocessing between design and simulation).

IT support

As mentioned earlier, the column “IT support” is related to the IT that is used in direct support of the listed tasks or in managing information related to the listed task or deliverable. This aspect raises potential for KBE if the listed IT support has potential to host the KBE application by e.g. offering scripting possibilities. Another way in which the IT support can contribute to a higher potential for KBE is through absence of typical IT barriers such as closed APIs or poor support for neutral information formats in those cases where KBE application would need to interact with such systems.

Reuse/consultation

The main difference between reuse and consultation from the perspective of KBE is that reuse implies that the knowledge originates from within the context where it is applied. This in turn is more beneficial in respect to the maintenance of the knowledge base in the KBE application, and thus increases the potential for a more sustainable KBE application. An exception to this, observed in Paper A and B, is that consultation of experts could either indicate potential for a new KBE application (mimicking the expert) or involve an existing KBE application that could be adapted and made available to the context of knowledge application (as was the case with one of the KBE applications in Paper B).

Knowledge categories

In this aspect one has to start distinguishing between different kinds of KBE applications, in order to be able to make a mapping between knowledge types and KBE applications. The discussion will be kept around two generic types of product and process: know-what and know-how. The reason for omitting know-why is because it deals with knowledge about the reasons for product and process design, which is not essential for the operation of a KBE application (but is necessary for the documentation of the KBE application). Know-who is also left out because KBE is based on a codification strategy. The distinction between knowledge regarding the product/process solution and information is not necessary since both need to be in place for KBE applications. The following combinations of knowledge types enable different kinds of KBE applications:

Product know-what + product know-how

This knowledge setup enables the automation of design or configuration. Typically the product know-what states the requirements and boundary conditions, while the product know-how is embedded in the form of rules. There is no need for process knowledge because a classic knowledge-based system can be used where the knowledge base (rules) is executed by an inference engine which generates the process at run time to find solutions which correspond to the requirements and satisfy the conditions. Examples are KBE applications numbers 2, 6 and 13 in Appendix A.

Product know-what + product know-how + process know-what + process know-how

This knowledge setup enables automation of design and evaluation, or even optimization as in the case of KBE application number 14 in Appendix A. It also enables the automation of design and configuration that does not utilize a knowledge-based system with an inference engine, but applies predefined process knowledge to the product knowledge through e.g. scripting, as exemplified by KBE application number 23 in Appendix A.

Product know-what + process know-what + process know-how

This knowledge setup enables automation of evaluation. An example can be a simulation that uses predefined process know-what and process know-how and later compares the simulation results with the requirements that need to be satisfied. An instance of this is found in Paper B (divided among two different applications, one that simulates and one that compares results with requirements). Another category that applies this knowledge combination consists of the kinds of automatic preprocessing which consider the details of the design. An example is meshing applications which apply different meshing rules (process know-how) depending on the sort of feature in the model or material (stated by product know-what).

Process know-what + process know-how

This knowledge setup enables the automation of administrative tasks such as information searching or preprocessing and analytic tasks such as simulation which are not product-dependent. The application in Paper C is a typical example of automation of the administrative process whose purpose was to collect and present the right information from the right sources in the right sequence.

Method and Level of formalization

Since KBE is based on codification and a high level of formalization, it is more beneficial if the current method for applying knowledge is based on codification than if a personalization approach is currently used or preferred.

Problems

A problem that can benefit a codification approach, and in turn also KBE, is that a loss of knowledge is perceived to take place, e.g. through key individuals leaving the organization for different reasons. Another problem that can benefit KBE in the same way is that a knowledge bottleneck (Guida and Tasso, 1995) is perceived to be in place, usually due to limitations associated with a personalization approach. The only way to increase the scale of the knowledge flow is through codification, which also raises the potential for KBE. The case in Paper C exemplifies such a situation, where the knowledge that previously was applied using manufacturing engineers in product development projects is replaced with a KBE application because the previous approach was considered a bottleneck.

5.6 Summary

This section describes a generic framework which can be applied to any group of engineers or to any process for which a knowledge management strategy is sought to be explicitly defined in terms of a plan of knowledge management actions devised towards business objectives. Its focus is on defining how and why knowledge is or should be applied in the product development process in order to guide how and why that knowledge should be captured. These discussions are guided and documented in the different knowledge profiles from which knowledge management actions can be derived and where each action is guided by a business objective. Combined with a neutral relation towards knowledge management solutions in terms of methods and IT, the approach has a heavy process focus to establish a well-founded image of the needs for knowledge management in the context of a specific process or organizational unit. The framework is based on the lessons learned from and a generalization of the approaches described in Papers D and E as well as in (Ćatić and Malmqvist, 2010b), with additions of important aspects found in the literature. It constitutes the synthesis of this thesis and has not been evaluated in the exact form proposed here. The next chapter provides a discussion of the results related to the posed research questions and literature, discussion of the contribution and a discussion of the research process.

"Follow effective action with quiet reflection. From the quiet reflection will come even more effective action."

[Peter Drucker]

6 Discussion

This chapter discusses the research results in relation to the research questions and the purpose of this thesis. Furthermore, the contributions and the area to which they belong are discussed and the chapter is rounded off with a reflection on the research setup and the research process.

6.1 Discussion of the research questions and results

The discussion of the research questions is primarily related to the purpose of the thesis, which is a wider utilization of KBE in product development. For each of the research questions the discussion is based on how an answer to the question contributes to this overall purpose.

> ***RQ1:*** *How can an explicit regard for the constraints posed by the PLM architecture in the development of KBE applications contribute to a wider utilization of KBE in product development?*

The underlying assumption of this research question is that a regard for architectural constraints can contribute to a wider utilization. This assumption rests partly on the results from the field of enterprise architecture, whose main area of concern is to make sure that the interaction between processes and IT is governed in such a way that the IT support for the processes is optimal. This is valid for any IT implementation and does not guarantee a wider use of IT but an optimal use of IT. With the results presented in Papers A, B and C it is argued that an explicit regard for architectural constraints contributes to a better utilization of KBE in product development. The utilization is better because the explicit regard for architectural constraints ensures that KBE is applied only where suitable and only to the extent that is required for the particular task, as made evident by the KBE application in Paper C. This in turn ensures that the benefit of the KBE application exceeds the cost. This reasoning does not, however guarantee a wider utilization of KBE. In fact it can render a narrower utilization of KBE if the business operations are such that KBE is not a suitable solution. Since product development is a knowledge-intensive process and the current utilization of KBE is low, there is however potential for a wider utilization. An answer to how the explicit regard for architectural constraints contributes to a wider utilization lies in the following examples of architectural principles from The Open Group Architecture Forum (TOGAF) (Schekkerman, 2003) standard for enterprise architectures:

- Maximize business benefit
- Common use of applications
- Data are accessible
- Data are shared

These principles are valid for any application of IT and, if followed, ensure that the costs involved with the development and maintenance of the IT application are minimized and the benefits maximized in a given context. Compared to other types of IT applications utilized in product development, a distinguishing characteristic of KBE from a PLM perspective is that, in order to address the aim of PLM to support the design of products which are optimized from a lifecycle perspective, it needs to access data from a larger variety and a larger amount of databases. If all the data needed are hard-coded, instead of accessed at their place of origin, the costs involved with the development and maintenance of the KBE application would rise substantially. For example, if the data needed for the demonstrator in Paper C had been hard-coded into the application, instead of accessed from the manufacturing databases, this would have rendered high costs partly because the data were dispersed across several sources with different formats and partly because the data was constantly changing with the manufacturing process. Similar thinking can be applied to functions performed by different applications. If there is a PDM system which already performs configuration of digital mock-ups, there is no financial soundness developing this function again as part of a KBE application. The effect of lower investment and higher benefits is directly translated into lower barriers and shorter lead time from the point at which an idea for a KBE application is born until it is approved and implemented, which in turn caters for a wider utilization of KBE.

> ***RQ2:*** *How can the concept of a service oriented PLM architecture enable an explicit regard for the PLM architecture in the implementation of KBE applications?*

The previously stated architectural principles related to use of applications and data access and sharing can be reformulated in the following two principles for the IT environment. In the IT environment there should be:

1. No redundancy of functions
2. No redundancy of data

The first principle drives a need for modularization (as elaborated in the concept of knowledge modules in Paper A and demonstrated in Paper B) implying that the complete IT environment should be made up of functional blocks which can be combined in different ways. The second principle (and to some extent also the first one) drives the need to focus on interfaces and communication between the elements inside the IT environment. The second principle was in strong focus in Paper C.

Both of these principles are fundamentally related to the aims and purposes of a service-oriented architecture. An answer to Research Question 2 is that the service-oriented architecture enables an explicit regard for the constraints of the PLM architecture essentially by allowing the KBE developer and implementer to more easily access other functions and data. It is, however, not only the access to the functions and the data that is enabled by an SOA. A properly documented SOA makes it easier to identify also which systems the KBE application needs to access for functions and data.

The effect of this, as observed in both Paper B and Paper C, is that the effort needed to implement a new or re-implement an existing KBE application in a new task is lower, due to the need to develop less dedicated code and implement only one new interface in the KBE application (the interface towards the service bus). The wider implication of this is that a recognized business process need can be more quickly satisfied, and the product development process can be improved continuously with many small changes and adaptations as the business environment changes. The contribution of Research Question 2 and its answer to the overall purpose is provided by the fact that the service-oriented architecture facilitates an explicit regard for architectural constraints and principles, which in turn contribute to a wider utilization as previously discussed.

> ***RQ3:*** *How can explicit knowledge management strategies in product development be formulated and methodologically supported?*

This research question reflects the need to perceive KBE not only as a method for process improvement but also as a method for knowledge management. As such it needs to comply with the knowledge management strategy in the process or organization unit in question. In order to investigate whether KBE does comply with the knowledge management strategy, the strategy itself has to be explicitly formulated, and an answer to this research question guides one in doing so. It is therefore argued here that by enabling the formulation of knowledge management strategies, the investigation of KBE as a suitable method is also enabled (the investigation itself is supported by the answer to Research Question 4). By making such an investigation possible, this contributes to the overall purpose of the thesis, since the ability to identify potential for KBE is a key factor for increasing the ability to utilize KBE more widely in product development.

Based on the framework described in Section 5, and thus implicitly based on the experiences from the appended papers, the inclusion of the following aspects is a fundamental part of an answer to how a knowledge management strategy can be formulated and methodologically supported:

1. **Bottom-up approach**
   The types of knowledge important for the processes vary between organizational units and processes, because their tasks and deliverables call for an accentuation of different types of knowledge. For this reason, and in order to increase the motivation to engage in a knowledge management strategy, the formulation of the strategy has to start from the needs of those who are supposed to engage in it. This conveys a clear message that the knowledge management strategy is put in place as a way to better support those involved

in the business operations to achieve the strategic objectives and continuously improve both the products and the processes.

2. **Start from knowledge application**
   Since the desired effects of knowledge management are realized only if the knowledge is reused, then this should also be the starting point to answer the question "which knowledge is most effective when applied".
3. **Focus on knowledge in relation to process tasks and process deliverables**
   This point constitutes an extension of the previous point, and highlights the need to focus the discussion and not talk about knowledge in general. This focus is important in order to be able to distinguish between knowledge that is important and knowledge that is "generally interesting" but of limited value for the process.

RQ4: *Which attributes characterize a knowledge management strategy in product development for which KBE is feasible as methodological support?*

The results related to this research question are summarized in Section 5.5 and indicate that no clear-cut "mapping" of attributes can be defined that guarantees a high potential for KBE. The question itself is posed to complement the existing methodological frameworks related to KBE, which try to look at the process and the product to identify potential for KBE. Some general categories of attributes which form an answer can be distinguished from the results:

- **Existence of appropriate types of knowledge**
  It was elaborated in Section 5.5 which types of knowledge are related to different types of KBE applications.
- **High level of formalization and maturity**
  The existence of right knowledge types is an important prerequisite but it is not enough. Just as implied by both Sunnersjö (1994) and Pugh (1996), the knowledge has to have a certain level of formalization and maturity in order for KBE to be successfully applied and maintained. The consequence of having low maturity is not necessarily noticed during the development and implementation of a KBE application, but during maintenance. Since low maturity means that knowledge is likely to change, it directly affects the maintenance effort related to the knowledge base. If the KBE application is such that the knowledge base is easily changed, the negative effects of low maturity are smaller than if the KBE application contains product models and hard-coded process sequences where even small changes require a large programming effort to implement.
- **Codification strategy**
  The fact that KBE is based on codification means that this approach also has to be utilized in the knowledge management strategy, at least for those knowledge types that are needed for the KBE application. If the identified knowledge is highly tacit, then resources have to be invested to externalize the knowledge needed for the development and implementation. Resources have to be invested in developing methods or even in

educating the individuals whose knowledge is used to further develop the knowledge base in the KBE application.

Since KBE ultimately rests on codification and automation, the knowledge management strategy has to be based on a bottom-up approach and surrounded with transparency. The KBE application has to be accepted and perceived as a necessary solution to reach business objectives, instead of being perceived as a competitor that is supposed to replace people. In order for this to happen, the users and those whose knowledge is embedded in the KBE application have to agree on the role and the purpose of the KBE application. Such an agreement can only be reached with transparency surrounding all the aspects of the KBE application, and the proposed EFFEKT framework is a way of attaining this transparency in a structured way.

It can also be discussed whether Research Question 4 has a general answer that can be more concrete than a general list of categories related to knowledge management provided here and of categories related to the product and the process provided within the other frameworks in Section 2.1.1. The question tries to balance between a pull perspective, i.e. the knowledge management strategy which constitutes the source of needs and requirements, and a push perspective, i.e. the KBE methodology as a solution of some kind. The contribution of this research question to the overall purpose is that its answer contributes to the ability to identify potential KBE applications – which, as mentioned earlier, is a key factor for wider utilization of KBE in product development.

6.2 Discussion of EFFEKT from a knowledge management perspective

This section discusses how the aspects of knowledge transfer and motivation (described in Section 2.2.3) are reflected in the design and application of EFFEKT.

6.2.1 Knowledge transfer aspects in EFFEKT

Knowledge capture and application always entail knowledge transfer across time, and sometimes also across one or both of the dimensions of context and individuals, illustrated in Figure 2-9 in Section 2.2.3. The mode in which knowledge is transferred has to be considered in the choice of methods and IT support, mainly in knowledge capture. The following points highlight in which way knowledge transfer modes affect the choice of approach supported with methods and IT.

Transfer across context

A transfer across context requires that the knowledge is generalized to an appropriate level so that it provides support when applied in another context. An example from product development is given by Rinman and Wilson (2010) in their study of a lessons-learned implementation. They noted that knowledge captured in the lessons-learned documents was considered inapplicable because it was highly contextualized. In other words, the knowledge reflected very project-specific situations and the likelihood of those situations occurring again was considered very low. Rinman and Wilson (2010) further found that the methodological support for knowledge capture was poor. It did not ensure that the knowledge was made general enough to be reusable across the differing contexts of different projects. The generalization of knowledge can be addressed through methodological support, i.e. the method for knowledge capture guides the knowledge producer and intermediary to cater for this aspect. It can, however, also be addressed through process support by having a "knowledge release process" where someone besides the knowledge producer and intermediary reviews the knowledge and gives feedback regarding its generality.

Transfer across individuals

If a personalization approach is chosen between capture and application, it is important for the knowledge transfer processes that the individuals are able to find each other. This may sound obvious, but if left unaddressed it can cause issues in practice. In Paper D some of the component designers stated that the design rationale (the know-why) regarding the mechatronic subsystem architecture was lacking in the documentation that was provided from the R&I division, and that it was impossible for them to find the person who could provide them with the design rationale. This was because the technology transfer was not accompanied by an explicit strategy for knowledge transfer (other than a move of a handful of individuals from R&I to Driveline, which was assumed to implicitly cater for the knowledge transfer but apparently was not sufficient from the perspective of Driveline designers). A similar kind of reasoning can be applied also if a codification approach is selected, with the difference that it translates to making the knowledge easy to find, which in turn translates to how the knowledge is stored and structured.

6.2.2 Motivational aspects in EFFEKT

Since the effects of a knowledge management strategy can be realized only if the strategy is implemented, the motivational aspect (especially of the individuals) is very important to address. The motivational barriers described in Section 2.2.3 are addressed in different ways through different elements of the proposed framework.

The "lack of time" barrier is considered to be mostly a managerial issue and is addressed mainly by the "process integration" elements. These elements are there to make sure that the activities related to knowledge application/capture/development are given a sufficient amount of time, and that their execution is not only fully supported and financed but also expected. The feedback from users in (Ćatić and Malmqvist, 2010b) clearly demonstrates that this kind of managerial commitment is expected by the people working in the organization.

The "lack of knowledge" states that primarily the knowledge producers lack the knowledge of how to formulate and capture their own knowledge. This is addressed in the "method and IT" element and in the specification of the "to-be knowledge capture profile", which stimulates a discussion about, and provides transparency in, the process of selecting methods and IT which are appropriate for managing the particular types of knowledge in question. In addition, there exist motivational barriers related to the knowledge gap between the knowledge producers and consumers. There is either too much knowledge (yielding a need to “unlearn”) or too little knowledge (resulting in a poor understanding of what the new knowledge is useful for) in the receiving end (Cummings and Bing-Sheng, 2003). By iterating between and considering the context where knowledge is applied and the context where it is captured, there is a chance to identify whether there is a risk of the knowledge gap being either too small or too large, and address it.

The “lack of means” barrier is directly related to the support chosen for knowledge capture/application/development. This barrier is explicitly addressed through the element of “methods and IT” along with the different knowledge profiles, to make sure that appropriate methods and equipment are in place with respect to the types of knowledge in question.

Finally, the “lack of will” barrier is not directly related to anything specific. Dixon (2000) states that it can be attributed to one of the previous lacks, while Markus (2001) states that it is addressed through appropriate incentives. In the framework, the lack of will is primarily addressed by the fact that knowledge management is made as explicit as possible in order to create transparency and offset any assumptions to why certain knowledge is managed, or why certain methods and IT are chosen for its management. The transparency is achieved by connecting the management of important types of knowledge to the business objectives, to make it clear why certain types of knowledge are managed. Transparency is also created through the documentation of the knowledge profiles, to show clearly why certain methods and IT are chosen. Finally, transparency is achieved by showing how knowledge capture, application and development are interconnected, to make it clear which are the benefits of managing the knowledge and who is benefited in order to make it possible to discuss a rearrangement of time and resources from knowledge application towards knowledge capture.

6.3 Discussion of the contribution

The main areas of contribution by this thesis are product development research and practice. The results presented in Chapters 4 and 5 reflect the fact that this is an area which applies theories, models and results from other areas to the context of product development with the ultimate aim of improving product development research and practice. The areas that have been applied in this thesis are depicted in Figure 2-1. More concretely it is argued that the contributions of this thesis are within the sub-topics of KBE development methodology and knowledge management in product development. Even more concretely, the contributions are:

- The recognition that explicit regard for the constraints of the PLM architecture in KBE development and implementation contributes to a wider utilization of KBE in product development (Paper A).
- A demonstration of the effects of an explicit regard for the PLM architecture constraints in KBE development (Paper C).

- A demonstration of service-oriented architecture as an enabler for the explicit regard for architectural constraints (Paper B).
- The recognition that KBE needs to comply with the knowledge management strategy in the process/organizational unit in which it is applied (this thesis).
- A demonstration of how the needs for knowledge management in a product development process/organization can be identified and methodologically supported (Papers D and E).
- A framework for formulating explicit knowledge management strategies in product development with support for identifying potential for KBE applications (Chapter 5 in this thesis).

Taking the perspectives of generalizability and transferability of the results, some additional aspects of the contributions can be discussed. The transferability of the demonstrators in Paper B and C is low because they reflect unique needs and characteristics found in each of those cases. The recommendations in Paper D along with the method in Paper E have a higher transferability, but only to cases which have contexts similar to those described in the papers. The generalizability of the results is, however, high and a generalization is embodied in the listed contributions. The results from Papers A to C have been generalized in relation to the topic of regard for PLM architecture in KBE development and implementation, and the results from Papers D and E have been generalized in relation to the topic of knowledge management in product development. As a result, the transferability of the listed contributions is high, since they are in no way dependent on engineering domain, business domain or company size.

6.4 Discussion of the research setup and process

As described in Chapter 3 this research has been carried out in close collaboration with an industrial partner. Though this is positive since the area of contribution is product development research and practice, it has posed some challenges in managing the research process in a way that is supported by the industrial partner and, at the same time, focused on the exploration of new knowledge. The main challenge has been to maintain the balance between academic interests and industrial interests throughout the research process. While it is hard to isolate oneself totally from possible political agendas among the different industrial representatives, who may seek to influence the research process in a direction that favors their particular position, the research process has followed a direction mainly set by academic interests, indicating that a proper balance was maintained. An important factor in maintaining this balance has been wide support from Global Corp. which, along with the large size of Global Corp., resulted in the research project having no problems in finding suitable cases for the different studies and with individuals interested in participating. One exception to this, however, was the unwillingness of the IT system owners in Paper C to provide access to the data needed in the demonstrator.

One specific event that occurred during the research process deserves some extra attention – the change in focus that occurred between Paper C and Paper D. The change was based on my conclusion that the lack of knowledge management perspectives on KBE deserved more attention. During the studies in Papers A to C, the knowledge management perspective of KBE was highly relevant among, and questioned by, the different practitioners involved in the studies, while the body of knowledge related to KBE did not reflect this perspective as much. The fact that the change of focus occurred without the need for any substantial changes in the research setup between Project I and Project II reflects the previous statement that academic interests had the main influence on the direction of the research process.

6.5 Summary

This chapter has discussed the results presented in Chapters 4 and 5 and the way these address the research questions, as well as how they contribute to the overall purpose of a wider utilization of KBE in product development. Furthermore, a list of contributions is presented and the main area of contribution is claimed to be product development research and practice in general and KBE development methodology in particular and the contributions are held to be both general and transferable within the area of contribution. Finally, the research setup and process are discussed with the main conclusion that a proper balance between industrial and academic interests has been maintained throughout the research process.

"Reasoning draws a conclusion, but does not make the conclusion certain, unless the mind discovers it by the path of experience."

[Roger Bacon]

7 Conclusions

The application of knowledge-based engineering (KBE) has been demonstrated to possess high potential in automating tedious and time-consuming tasks, thus leaving more time and resources for creativity and innovation in product development. A wide utilization of KBE, however, has not yet occurred in product development practice. This thesis has adopted the perspective that a wider utilization can be achieved by an increased integration of KBE and PLM and by an increased ability to identify potential for KBE applications. Existing methodological frameworks for supporting the development of KBE applications focus more on internal functions and problem-solving techniques, and to a lesser extent on the fact that the KBE application is going to be part of a process and IT environment, i.e. the PLM architecture, with which it needs to comply. In addition, the existing frameworks and applied approaches for development and implementation of KBE tend only to adopt an IT perspective of KBE neglecting the fact that it is also a knowledge management method.

This thesis argues that there is a need to expand the existing methodologies with an explicit regard for the architectural adherence of the KBE application, in order to achieve a better integration of KBE with PLM and thus contribute to a wider utilization of KBE in product development. The thesis has investigated the effects of an explicit regard for the constraints of the PLM architecture in KBE development. The conclusion is that, besides a better integration with PLM, KBE applications can be developed with less programming effort and with maintained functionality. A service-oriented architecture has been investigated (SOA) as an enabler of explicit regard for the PLM architecture in the development and implementation of KBE. It is concluded that the SOA provides the KBE developer with easier access to data and functions which are requested from other parts of the PLM architecture. The KBE developer also needs to invest less effort in making and later maintaining interfaces, because only the interface towards the service bus is necessary. Furthermore, the SOA enables the reuse of existing KBE applications in different processes, thus benefiting a wider utilization of them.

This thesis also argues that complementing the process and IT perspective of KBE with a knowledge management perspective can benefit the existing methodologies by strengthening their ability to identify potential for KBE applications. The thesis has explored the area of knowledge management in product development and identified a need for methodological support for formulating explicit knowledge management strategies in product development which is addressed with "EFFEKT", an Expedient Framework for Formulating Explicit

Knowledge management sTrategies in product development. EFFEKT is based on the experiences from the development of methodological and IT support for knowledge management in Papers D and E and in (Ćatić and Malmqvist, 2010b), complemented with knowledge management models and theories from literature. EFFEKT highlights that the formulation of an explicit knowledge management strategy is driven by business objectives which call for application of certain types of knowledge in the business processes, determined by the processes and the organization under consideration. The specific mix of knowledge types (called knowledge profiles) present in a certain process and/or organization can, in turn, be used to identify methodological and IT solutions suitable for managing the knowledge.

Finally a discussion regarding which characteristics of an explicit knowledge management strategy, formulated using EFFEKT, that indicate potential for KBE applications yielded the following points:

- **The existence of appropriate knowledge types**
  The knowledge categorization used is the division into product and process knowledge, with further division into know-what, know-why and know-how. Different types of KBE applications require the existence of different combinations of these categories in the elements that are planned to be used as knowledge sources for the KBE application. The details are elaborated in Section 5.5.
- **High formalization and maturity of knowledge**
  In order for the knowledge base to be stable, or even formulated in a computer-executable form, the level of formalization and the maturity of the knowledge need to be high. Put simply, there must be a clear set of validated and verified knowledge (e.g. rules or constraints) in order for KBE to be a realistic solution.
- **Codification strategy**
  The organization has to accept that codification of knowledge is a valid strategy for managing the types of knowledge in question.

"I try to learn from the past, but I plan for the future by focusing exclusively on the present. That's where the fun is.

[Donald Trump]

8 Future work

The areas for future work primarily focus on the EFFEKT framework described in Chapter 5. There is a need to evaluate the framework in order to validate its general approach and design, by implementing it according to the described process applying the framework in its entirety rather than focusing only certain parts. One aspect in the framework that should be focused upon is the "type of knowledge" column, which is quite central in the framework since it is used to provide the traceability between the contexts of knowledge application and knowledge capture. Also the aspect of knowledge categorization is interesting for closer investigation, since the categorization of product/process know-what-how-why might still be rather abstract to support a practitioner in a concrete situation. A further concretization of this categorization would also be beneficial for enabling a more precise prescription of methods and IT suitable for different strategies combined with knowledge categories. Preferably the context for evaluative implementations of the framework should be outside the automotive industry in order to address its transferability to other businesses. The evaluation and validation of the framework rest on the extent to which its application leads to knowledge management solutions which themselves are valid in their context.

Future work which is directed more towards the area of KBE in the context of the proposed knowledge management framework relates to the need to investigate and further concretize the characteristics of a knowledge management strategy in which KBE is a suitable solution. The list of aspects provided in Section 5.5 is in no way exhaustive, and each of the aspects can be further elaborated and detailed. A possible approach for such a study is to analyze KBE applications which are successfully implemented in practice from a knowledge management perspective, and assess the details of the aspects listed in Section 5.5 for each KBE application to see if any patterns or relations can be identified.

Future work related to KBE from a process and IT perspective includes the need to further validate the anticipated effects of an SOA on KBE development and implementation. It is primarily the KBE lifecycle aspects of easier maintenance and easier reuse of KBE applications that are referred to. Such a study would require a company which has implemented an SOA and then implemented a KBE developed with a regard for architectural constraints, which the study could follow through its life.

References

Ackoff, R. L. (1989) *From Knowledge to Wisdom*, **Journal of Applied Systems Analysis**, Vol. 16, No. pp. 3-9

Ahmed, S. (2005) *Encouraging Reuse of Design Knowledge: a method to index knowledge*, **Design Studies**, Vol. 26, No. 6, pp. 565-592

Ahmed, S., Kim, S. and Wallace, K. (2005). *A methodology for creating ontologies for engineering design*, **2005 ASME DETC Design, Theory and Methodology Conference**, Long Beach, California, USA, Paper ID: DETC2005-84729

Andersson, P., Wolgast, A. and Isaksson, O. (2008) *Current Industrial Practices for Re-use of Manufacturing Experience in a Multidisciplinary Design Perspective*, **Proceedings of 10th International Design Conference - DESIGN 2008**, Dubrovnik, Croatia pp. 885-892

Andreasen, M. M. and Hein, L. (1987) *Integrated Product Development*, **The Institute for Product Development, Technical University of Denmark**, Lyngby, Denmark

Bachrach, H. E. (1997) *Formal Methods for Design Automation Application Development*, Master Thesis, **Massachusetts Institute of Technology**, Boston, MA,

Bergsjö, D., Ćatić, A. and Malmqvist, J. (2008) *Implementing a Service-oriented PLM Architecture Focusing on Support for Engineering Change Management*, **International Journal of Product Lifecycle Management**, Vol. 3, No. 4, pp. 335-355

Bergsjö, D., Malmqvist, J. and Ström, M. (2006) *Architectures For Mechatronic Product Data Integration in PLM Systems*, **Proceedings of 9th International Design Conference - DESIGN 2006**, Dubrovnik, Croatia pp. 1065-1076

Bermell-Garcia, P. and Fan, I.-S. (2008) *Practitioner requirements for integrated Knowledge-Based Engineering in Product Lifecycle Management*, **International Journal of Product Lifecycle Management**, Vol. 3, No. 1, pp. 3-20

Blessing, L. T. M. and Chakrabarti, A. (2009) *DRM, A Design Research Methodology*, **Springer-Verlag**, London, UK

Blount, G. N. and Kneebone, S. (1995) *Selection of knowledge-based engineering design applications*, **Journal of Engineering Design**, Vol. 6, No. 1, pp. 31-38

Boart, P., Andersson, P. and B-O, E. (2006) *Knowledge Enabled Pre-processing for Structural Analysis*, **Proceedings of NordPLM 06**, Göteborg, Sweden pp. 89-97

Bracewell, R. H., Ahmed, S. and Wallace, K. M. (2004). *DRED and Design Folders, A way of Capturing, Storing and Passing on, Knowledge Generated During Design Projects*, **Proceedings of ASME 2004 DETC Computers and Information in Engineering Conference**, Salt Lake City, Utah, USA, Paper ID: DETC2004-57165

Brissaud, D., Garro, O. and Poveda, O. (2003) *Design Process Rationale Capture and Support by Abstraction of Criteria*, **Research in Engineering Design**, Vol. 14, No. 3, pp. 162-172

Burr, H., Vielhaber, M., Deubel, T., Weber, C. and Haasis, S. (2005) *CAx/Engineering Data Management Integration: Enabler for Methodical Benefits In the Design Process.*, **Journal of Engineering Design**, Vol. 16, No. 4, pp. 385-398

Ćatić, A. (2009) *Integrating KBE and PLM Through a Service Oriented Architecture*, Licentiate thesis, **Chalmers University of Technology**, Göteborg, Sweden,

Ćatić, A. and Andersson, P. (2008). *Manufacturing Experience in a Design Context Enabled by a Service Oriented PLM Architecture*, **2008 ASME DETC Design for Manufacturing and the Lifecycle Conference**, New York City, USA, Paper ID: DETC2008/49858

Ćatić, A. and Malmqvist, J. (2010b) *Implementing a Wiki to Capture and Share Engineering Knowledge*, **Proceedings of NordDesign 2010**, Göteborg, Sweden pp. 171-184

Ćatić, A. and Vielhaber, M. (2011) *Lean Product Development: Hype or Sustainable New Paradigm?*, **Proceedings of 18th International Conference on Engineering Design - ICED 11**, Copenhagen, Denmark pp. 157-168

Cederfeldt, M. (2004). *Variant Design Automation - Strategy and Procedure for Storing Design Process Information*, **2004 ASME DETC Computers and Information in Engineering**, Salt Lake City, Utah, USA, Paper ID: DETC2004-57377

Cederfeldt, M. (2007) *Planning Design Automation - A Structured Method and Supporting Tools*, Doctoral Thesis, **Chalmers University of Technology**, Göteborg, Sweden,

Cederfeldt, M. and Elgh, F. (2005). *Design Automation in SMEs - Current State, Potential, Need and Requirements*, **Internation Conference on Engineering Design ICED05**, Melbourne, Australia, Paper ID: 314.46

Cederfeldt, M. and Elgh, F. (2005). *Design Automation in SMEs - Current State, Potential, Need and Requirements*, **Proceedings of 15th International Conference on Engineering Design ICED '05**, Melbourne, Australia, Paper ID: 314.46

Chakrabarti, A., Kota, S., Rao, N. and Chowdary, S. (2005). *A Designer Workbench With Real-time Capture and Reuse of Information on Product Evolution and Rationale*, **International Conference on Engineering Design ICED05**, Melbourne, Australia, Paper ID: 267.91

CIMData (2002) *Product Lifecycle Management*, Retrieved on: 2008-12-12 from www.cimdata.com.

Clarkson, P. J. and Eckert, C., Eds. (2005). Design Process Improvement - A review of current practice. London, Springer Verlag London Limited.

Clarkson, P. J. and Hamilton, J. R. (2000) *'Signposting': a parameter-driven task-based model of the design process*, **Research in Engineering Design**, Vol. 12, No. 1, pp. 18-38

Colombo, G., Girotti, A. and Rovida, E. (2005). *Automatic Design of Hydraulic Press Brake for Sheet Metal Bending*, **International Conference on Engineering Design ICED05**, Melbourne, Australia, Paper ID: 9.39

Cooper, R. G. (2008) *Perspective: The Stage-Gate® Idea-to-Launch Process—Update, What's New, and NexGen Systems*, **Journal of Product Innovation Management**, Vol. 25, No. 3, pp. 213-232

Counsell, J., Porter, I., Dawson, D. and Duffy, M. (1999) *Scemebuilder: computer aided knowledge based design of mechatronic systems*, **Assembly Automation**, Vol. 19, No. 2, pp. 129-138

Cross, N. (1994) *Engineering Design Methods - Strategies for Product Design*, **John Wiley & Sons Ltd**, Chichester, UK

Cummings, J. L. and Bing-Sheng, T. (2003) *Transferring R&D knowledge: the key factors affecting knowledge transfer success*, **Journal of Engineering and Technology Management**, Vol. 20, No. 1-2, pp. 39–68

Dassault Systèmes (2011) *Product Lifecycle Management PLM Solutions*, Retrieved on: 2011-07-23 from www.3ds.com.

Davenport, T. H. and Short, J. E. (1990) *The New Industrial Engineering: Information Technology And Business Process Redesign*, **Sloan Management Review**, Vol. 31, No. 4, pp. 11-27

Dixon, N. (2000) *Common Knowledge: How Companies Thrive by Sharing What They Know*, **Harvard Business Press**, Boston, MA, USA

Dunning, J. H. (2002) *Regions, Globalization, and the Knowledge-Based Economy*, **Oxford University Press**, Oxford, UK

Elgh, F. and Cederfeldt, M. (2005). *A Design Automation System Supporting Design For Cost - Underlying Method, System Applicability and User Experiences*, **CE05: ISPE international Conference on Concurrent Engineering**, Fort Worth, Texas, USA,

Eppinger, S. D., Whitney, D. E., Smith, R. P. and Gebala, D. A. (1994) *A Model-Based Method for Organizing Tasks in Product Development*, **Research in Engineering Design**, Vol. 6, No. 1, pp. 1-13

Fu, Q. Y., Chui, Y. P. and Helander, M. G. (2006) *Knowledge Identification and Management in Product Design*, **Journal of Knowledge Management**, Vol. 10, No. 6, pp. 50-63

Fuxin, F. (2005) *Enabling Preconditions for Geometrical Application in Preconceptual Stages of Engineering Design*, **Proceedings of 15th International Conference on Engineering Design ICED '05**, Melbourne, Australia pp. 79-80 (executive summary)

Gammelgaard, J. and Ritter, T. (2005) *The Knowledge Retrieval Matrix: Codification and Personification as Separate Strategies*, **Journal of Knowledge Management**, Vol. 9, No. 4, pp. 133-143

Gardoni, M. and Dudezert, A. (2005). *Valuing Knowledge Management Impact on Engineering Design*, **International Conference on Engineering Design ICED05**, Melbourne, Australia, Paper ID: 171.33

Greg, H., McAlpine, H., Camarero, R., Culley, S. J., Leblanc, T. and Fortin, C. (2009) *The Management of Digital Skecthes through PLM solutions*, **Proceedings of 17th International Conference on Engineering Design ICED '09**, Stanford University, Palo Alto, CA, USA pp. 239-250

Guida, G. and Tasso, C. (1995) *Design and development of knowledge-based systems: from life cycle to methodology*, **John Wiley & Sons, Inc.**, New York, USA

Haas, R. and Sinha, M. (2004) *Concurrent engineering at Airbus - A case study*, **International Journal of Manufacturing Technology and Management**, Vol. 6, No. 3-4, pp. 241-243

Hansen, M. T., Nohria, N. and Tierney, T. (1999) *What's Your Strategy For Managing Knowledge*, **Harvard Business Review**, Vol. No. March-April, pp. 106-116

Harmon, P. (2004) *Enterprise Architectures*, **Business Process Trends**, Vol. 2, Nr. 1, pp.1-14, Retrieved on: 2008-12-11 from http://www.bptrends.com/publicationfiles/12%2D03%20NL%20Enterprise%20Architectures%2Epdf

Hoegl, M. and Schulze, A. (2005) *How to Support Knowledge Creation in New Product Development: An Investigation of Knowledge Management Methods*, **European Management Journal**, Vol. 23, No. 3, pp. 236-273

Hopgood, A. A. (2001) *Intelligent Systems for Engineers and Scientists*, **CRC Press LLC**, Boca Raton, Florida, USA

Huang, G. Q. (1996) *Design for X - Concurrent Engineering Imperatives*, **Chapman & Hall**, London, UK

Huang, G. Q. and Mak, K. L. (1998) *Compuiter aids for engineering change control*, **Journal of Materials Processing Technology**, Vol. 76, No. 1-3, pp. 187-191

Hubka, V. and Eder, W. E. (1996) *Design Science: Introduction to the Needs, Scope and Organization of Engineering Design Knowledge*, **Springer-Verlag Ltd**, London, UK

Hurwitz, J., Bloor, R., Baroudi, C. and Kaufman, M. (2007) *Service Oriented Architecture For Dummies*, **John Wiley & Sons**, Hoboken, New Jersey, USA

Hvam, L., Mortensen, N. H. and Riis, J. (2008) *Product Customization*, **Springer-Verlag**, Berlin, Germany

Isaksson, O., Seppälä, S. and Eppinger, S. D. (2000) *Evaluation of design process alternatives using signal* *ow graphs*, **Journal of Engineering Design**, Vol. 11, No. 3, pp. 211-224

ISO (2010) *International Organization for Standardization*, Retrieved on: 2010-12-13 from http://www.iso.org.

Jackson, P. (1999) *Introduction to Expert System - third edition,* **Addison-Wesley**, Harlow, UK

Jarratt, T. A. W., Eckert, C. M., Caldwell, N. H. M. and Clarkson, P. J. (2011) *Engineering change: an overview and perspective on the literature*, **Research in Engineering Design**, Vol. 22, No. 2, pp. 103-124

Johansson, J. (2007). *A Flexible Design Automation System for Toolsets for the Rotary Draw Bending of Aluminium Tubes,* **2007 ASME DETC - Computers and Information in Engineering** Las Vegas, Nevada, USA, Paper ID: DETC2007-34310

Johansson, J. (2008). *Manufacturability Analysis Using Integrated KBE, CAD and FEM,* **2008 ASME DETC - Computers and Information in Engineering** Brooklyn, New York, USA, Paper ID: DETC2008-49019

Johansson, J. and Sunnersjö, S. (2006). *Automated Design of Rotary Draw Bending Tools - An Approach Based on Generic CAD-Models Driven by Heuristic and Algorithmic Knowledge*, **Efficiency Development of Manufacturing Machines and Processes EDMMP 2006**, Wroclaw, Poland,

Joshi, N., Ameri, F. and Dutta, D. (2005). *Systematic Decision Support for Engineeering Change Management in PLM,* **Proceedings of DETC/CIE** Long Beach, CA, USA,

Kennedy, M., Harmon, K. and Minnock, E. (2008) *Ready, Set, Dominate: Implement Toyota's set-based learning for developing products and nobody can catch you,* **Oaklea Press**, Richmond, VA, USA

Kochan, A. (1999) *Jaguar Uses Knowledge Based Tools to Reduce Model Development Time*, **Assembly Automation** Vol. 19, No. 2, pp. 114-117

La Rocca, G., Krakers, L. and van Tooren, M. J. L. (2002). *Development of an ICAD Generative Model for Blended Wing Body Aircraft Design*, **9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization**, Atlanta, Georgia, USA,

Lee, H. J., Ahn, H. J., Kim, J. W. and Park, S. J. (2006) *Capturing and reusing knowledge in engineering change management: A case of automobile development*, **Journal of Information Systems Frontier**, Vol. 8, No. 5, pp. 375-394

Lee, Y. H. and Li, C. G. (2009) *Automation in the design of EDM electrodes* **Computer-Aided Design**, Vol. 41, No. 9, pp. 600-613

Li, C. L., Li, C. G. and Mok, A. C. K. (2005) *Automatic layout design of plastic injection mould cooling system*, **Computer-Aided Design**, Vol. 37, No. 7, pp. 645-662

Liebowitz, J. (1999) *Key Ingredients to the Success of an Organization's Knowledge Management Strategy*, **Journal of Knowledge and Process Management**, Vol. 6, No. 1, pp. 37-40

Liker, J. K. (2004) *The Toyota way: 14 management principles from the world's greatest manufacturer*, **McGraw-Hill**, New York, USA

Lundvall, B. and Johnson, B. (1994) *The learning economy*, **Journal of Industry Studies**, Vol. 1, No. 2, pp. 23-42

Malmqvist, J. (2008) *Product Lifecycle Management - Course Binder*, **Chalmers University of Technology**, Göteborg, Sweden

March, J. G. (1991) *Exploration and Exploitation in Organizational Learning*, **Organization Science**, Vol. 2, No. 1, pp. 71-87

Markus, M. L. (2001) *Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Situations and Factors in Reuse Success*, **Journal of Management Information Systems**, Vol. 18, No. 1, pp. 57-93

Mathworks (2008) *MATLAB and Simulink for Technical Computing*, Retrieved on: 2008-12-11 from www.mathworks.com.

Maxwell, J. A. (2005) *Qualitative Research Design - An Interactive Approach, 2nd Edition*, **SAGE Publications**, Thousand Oaks, CA, USA

McMahon, C., Lowe, A. and Culley, S. (2004) *Knowledge Management in Engineering Design: Personalization and Codification*, **Journal of Engineering Design**, Vol. 15, No. 4, pp. 307-325

Merkel, M. and Schumacher, A. (2003) *An Automated Optimization Process for a CAE Driven Product Development*, **Journal of Mechanical Design**, Vol. 125, No. pp. 694-700

Merriam-Webster (2011) *Webster's Online Dictionary and Thesaurus*, Retrieved on: 2011-08-25 from http://www.merriam-webster.com.

Mesihovic, S. (2004) *On the Development and Sales-Delivery Process of Configurable Products*, Licentiate thesis, **Chalmers University of Technology**, Göteborg, Sweden,

Microsoft (2008) *Microsoft Office*, Retrieved on: 2008-12-11 from http://office.microsoft.com.

Mittal, S. and Frayman, F. (1989) *Towards a Generic Model of Configuration Tasks*, **Proceedings of the 11th IJCAI**, San Mateo, CA, USA pp. 1395-1401

MOKA (2000) *MOKA: A framework for structuring and representing engineering knowledge*, Retrieved on: 2011-02-09 from http://web1.eng.coventry.ac.uk/moka/default.htm.

Morgan, J. M. and Liker, J. K. (2006) *The Toyota Product Development System*, **Productivity Press**, New York, USA

MSC Software (2008) *MSC Software*, Retrieved on: 2008-12-11 from www.mscsoftware.com.

Mutafelija, B. and Stromberg, H. (2003) *Systematic Process Improvement Using ISO 9001: 2000 and CMMI*, **Artech House**, Norwood, MA, USA

NASA (2010) *NASA Lessons Learned Information System*, Retrieved on: 2010-08-05 from http://llis.nasa.gov/llis/search/home.jsp.

Nonaka, I. (1994) *Dynamic Theory of Organizational Knowledge Creation*, **Organization Science**, Vol. 5, No. pp. 14-39

Norell, M. (1992) *Stödmetoder och samverkan i produktutveckling (Advisory Tools and Co-operation in Product Development) (In Swedish)*, PhD Thesis, **Royal Institute of Technology**, Stockholm, Sweden,

OMG (2005) *KBE Services for PLM - Request for Proposal, OMG Document: dtc/2005-09-11*, Retrieved on: 2008-11-13 from http://www.omg.org/cgi-bin/doc?dtc/2005-9-11.

OMG (2007) *Product Lifecycle Management Services V2.0*, Retrieved on: 2008-12-12 from http://www.prostep.org/fileadmin/freie_downloads/Empfehlungen-Standards/OMG/OMG_PLM-Services_2.0_07-04-01.pdf.

OMG (2008) *Unified Modelling Language*, Retrieved from www.uml.org.

Pahl, G. and Beitz, W. (1996) *Engineering Design - A Systematic Approach*, **Springer-Verlag**, London, UK

Pavković, N., Marjanović, D., Hoić, M. and Rohde, D. (2009) *An Explorer Style Interface for Design Knowledge Indexing Based on Taxonomies*, **Proceedings of 17th International Conference on Engineering Design ICED '09**, Stanford University, Palo Alto, CA, USA pp. 65-74

Penoyer, J. A., Burnett, G., J., F. D. and Liou, S.-Y. (2000) *Knowledge Based Product Life Cycle Systems: Principles of Integration of KBE and C3P*, **Computer-Aided Design**, Vol. No. 32, pp. 311-320

Persse, J. R. (2001) *Implementing the Capability Maturity Model*, **John Wiley & Sons**, New York, NY, USA

Persson, J. and Stiborg, J. (2008) *Development of a service oriented engineering framework*, Master Thesis, **Chalmers University of Technology**, Göteborg, Sweden, Report No. 3491

Pikosz, P. and Malmqvist, J. (1998). *A Comparative Study of Engineering Change Management In Three Swedish Engineering Companies*, **1998 ASME Design Engineering Technical Conference**, Atlanta, GA, USA, Paper ID: DETC98/EIM-5684

Pikosz, P. and Malmqvist, J. A. (1998) *Comparative Study of Engineering Change Management in Three Swedish Engineering Companies*, **Proceedings of DETC'98**, Atlanta, GA, USA pp.

Pinfold, M. and Chapman, C. (1999) *The Application of Knowledge Based Engineering Techniques to the Finite Element Mesh Generation of an Automotive Body-in-white Structure*, **Journal of Engineering Design**, Vol. 10, No. 4, pp. 365-377

Poenisch, V. and Clark, A. (2006) *Elements of an Interface Standard for Knowledge Sources in Knowledge-Based Engineering*, **ASME Journal of Computing and Information Science in Engineering**, Vol. 6, No. 1, pp. 78-83

Polanyi, M. (1967) *The Tacit Dimension*, **Routledge & Kegan Paul Ltd**, London, UK

Prasad, B. (1997) *Concurrent Engineering Fundamentals*, **Prentice Hall PTR**, Upper Saddle River, New Jersey, USA

Pugh, S. (1990) *Total Design*, **Pearson Education Limited**, Harlow, UK

Pugh, S. (1996) *Creating Innovative Products Using Total Design*, **Addison-Wesley**, Reading, Massachusetts, USA

Raffaelia, R. and Germania, M. (2008) *Knowledge-based approach to flexible part design*, **Journal of Engineering Design**, Vol. 21, No. 1, pp. 7-29

Raudberget, D. (2010) *The Decision Process in Set-Based Concurrent Engineering - An Industrial Case Study*, **Proceedings of 11th International Design Conference - DESIGN 2010**, Dubrovnik, Croatia pp. 937-946

Revilla, E., Prieto, I. M. and Prado, B. R. (2010) *Knowledge Strategy: Its Relationship to Environmental Dynamism and Complexity in Product Development*, **Journal of Knowledge and Process Management**, Vol. 17, No. 1, pp. 36-47

Rinman, N. and Wilson, S. (2010) *Learning from Product Development Projects - An Approach to Increasing Product Development Capability Through Knowledge Creation adn Reuse*, Master Thesis, **Chalmers University of Technology**, Göteborg, Sweden, E2010:005

Roozenburg, N. F. M. and Eekels, J. (1995) *Product Design: Fundamentals and Methods*, **John Wiley & Sons Ltd**, Chichester, UK

Rosenfeld, L. W. (1995) *Solid Modelling and Knowledge-Based Engineering*, Handbook of Solid Modelling. D. E. LaCourse, **McGraw-Hill**, New York, USA,

Rubenstein-Montano, B., Liebowitz, J., Buchwalter, J., McCaw, D., Newman, B. and Rebeck, K. (2001) *A systems thinking framework for knowledge management*, **Decision Support Systems** Vol. 31, No. pp. 5–16

Salehi, V. and McMahon, C. (2009) *Action Research Into the Use of Parametric Associative CAD Systems in an Industrial Context*, **Proceedings of 17th International Conference on Engineering Design ICED '09**, Stanford University, Palo Alto, CA, USA pp. 133-144

Sandberg, J. and Targama, A. (2007) *Managing Understanding in Organizations*, **SAGE Publications**, London, UK

Sandberg, M., Boart, P. and Larsson, T. (2005) *Functional Product Life-cycle Simulation Model for Cost Estimation in Conceptual Design of Jet Engine Components*, **Journal of Concurrent Engineering: Research and Applications**, Vol. 13, No. 4, pp. 331-342

Sandberg, M., Johnsson, H. and Larsson, T. (2008) *Knowledge-based engineering in construction: the prefabricated timber housing case*, **Journal of Information Technology in Construction**, Vol. 13, No. pp. 408-420

Sandberg, M. and Marefat, M. M. (2006). *Combining Knowledge-based Engineering and Case-based Reasoning for Design and Manufacturing Iteration*, **4th International Conference on Manufacturing Research**, Liverpool, UK,

Sandberg, M., Åström, P., Larsson, T. and Näsström, M. (2005) *A Design Tool Integrating CAD and Virtual Manufacturing for Distorsion Assessment*, **Proceedings of 15th International Conference on Engineering Design ICED '05**, Melbourne, Australia pp. 628-630 (executive summary)

Schekkerman, J. (2003) *How to Survive In The Jungle of Enterprise Architecture Frameworks: Creating or Choosing An Enterprise Architecture Framework*, **Trafford Publishing**, Victoria, British Columbia, Canada

Schreiber, G., Akkermans, H., Shadbolt, N., de Hoog, R. and Anjewierden, A. (2000) *Knowledge Engineering and Management: The CommonKADS Methodology*, **The MIT Press**, Cambridge, Massachusetts, USA

Siemens PLM (2011) *Siemens PLM Software*, Retrieved on: 2011-07-23 from www.plm.automation.siemens.com.

Sobek, D. K., Ward, A. C. and Liker, J. K. (1999) *Toyota's Principle of Set-Based Concurrent Engineering*, **Sloan Management Review**, Vol. 40, No. 2 pp. 67-83

Soininen, T., Tiihonen, J., Männistö, T. and Sulonen, R. (1998) *Towards a general ontology of configuration*, **Artificial Intelligence for Engineering Design, Analysis and Manufacturing**, Vol. 12, No. 4, pp. 357-372

Sriram, R. D. (1997) *Intelligent Systems for Engineering*, **Springer-Verlag**, London, UK

Stokes, M. (2001) *Managing engineering knowledge: MOKA: methodology for knowledge based engineering applications*, **Professional Engineering Publishing Limited**, London, UK

Stolt, R. (2005). *A CAD-Integrated System for Automated Idealization of CAD-Models for Finite Element Analysis*, **2005 ASME DETC - Computers and Information in Engineering** Long Beach, California, USA, Paper ID: DETC2005-84287

Stolt, R. (2008). *An Automated Design and Evaluation System for Pressed and Sintered Powder Metallurgy Parts*, **MPIF/APMI 2008 World Congress on Powder Metallurgy and Particulate Materials**, Wacshington D.C., USA

Strinning, P. (1995) *A Computerized Trial and Error Technique for Impeller Design*, **The ITT Flygt Scientific Magazine** Vol. No. 3, pp. 19-27

Stumptner, M. (1997) *An Overview of Knowledge-based Configuration*, **AI Communications**, Vol. 10, No. 2, pp. 111-125

Styhre, A. (2003) *Understanding Knowledge Management - Critical and Postmodern Perspectives*, **Liber Ekonomi AB**, Malmö, Sweden

Suh, N. P. (1990) *The Principles of Design* **Oxford University Press**, New York, USA

Sunnersjö, S. (1994) *AID-Features: A New Tool for Design Automation*, **Proceedings of CACD94: Lancaster International Workshop on Engineering Design**, Lancaster University Engineering Design Centre, Lancaster, Great Britain pp. 241-258

Svensson, D. (2003) *Towards Product Structure Management in Heterogeneous Environments*, Doctoral Thesis, **Chalmers University of Technology**, Göteborg,

Tamminenia, S. V., Raoa, A. R., Scanlanb, J. P., Reedb, P. A. S. and Keaneb, A. J. (2009) *A knowledge-based system for cost modelling of aircraft gas turbines*, **Journal of Engineering Design**, Vol. 20, No. 3, pp. 289-305

Thomas, M. and McGarry, F. (1994) *Top-Down vs. Bottom-Up Process Improvement*, **IEEE Software**, Vol. 11, No. 4, pp. 12-13

Tidstam, A. and Malmqvist, J. (2011) *Authoring and Verification of Vehicle Configuration Rules*, **International Conference on Product Lifecycle Management PLM11**, Eindhoven, Netherlands pp.

Ullman, D. G. (2003) *The Mechanical Design Process*, **McGraw Hill**, New York, USA

Ulrich, K. T. and Eppinger, S. D. (2008) *Product Design and Development (fourth edition)*, **McGraw Hill Inc**, New York, USA

Wallace, K., Ahmed, S. and Bracewell, R. (2005) *Engineering knowledge management*, Design Process Improvement - A review of current practice. P. J. Clarkson and C. Eckert, **Springer Verlag London Limited**, London, UK, pp. 327-343

Ward, A. C. (2007) *Lean Product and Process Development*, **Lean Enterprise Institute**, Cambridge, MA, USA

VDI 2221 (1993) *VDI guideline 2221: "Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte"*, **VDI-Verlag**, Düsseldorf, Germany

Weilguny, L. J. H. and Gerhard, D. (2009) *Implementation of a Knowledge-Based Engineering Concept at an Automotive OEM*, **Proceedings of 17th International Conference on Engineering Design ICED '09**, Stanford University, Palo Alto, USA pp. 69-80

Whitney, D. E., Dong, Q., Judson, J. and Mascoli, G. (1999). *Introducing Knowledge-Based Engineering Into an Interconnected Product Development Process*, **1999 ASME DETC Design, Theory and Methodology Conference**, Las Vegas, USA, Paper ID: DETC1999/DTM-8741

Williamson, K. (2002) *Research methods for students, academics and professionals: Information management and systems - 2nd edition*, **Centre for Information Studies**, Wagga Wagga, New South Wales, Australia

Wojciech, S. (2007) *Application of MOKA methodology in generative model creation using CATIA*, **Engineering Applications of Artificial Intelligence**, Vol. 20, No. 5, pp. 677-690

Zachman, J. A. (1987) *A Framework for Information Systems Architecture*, **IBM Systems Journal**, Vol. 26, No. 3, pp. 276-292

Zimmerman, T. (2008) *Implementing PLM Across Organisations - For Multi-disciplinary and Cross-functional Product Development*, Doctoral Thesis, **Chalmers University of Technology**, Göteborg, Sweden,

Zimmerman, T., Malmqvist, J. and Ćatić, A. (2008) *Implementing PLM In Practice - Findings From An Empirical Study In the Commercial Vehicle Industry*, **Submitted to Research in Engineering Design**,

Zmud, R. W. (1984) *An Examination of 'Push-Pull' Theory Applied to Process Innovation in Knowledge Work*, **Management Science**, Vol. 30, No. 6, pp. 727-738

Appendix A

A summary of the 24 KBE applications identified in the literature is given here.

Index of KBE applications:

1. CoRPP (Elgh and Cederfeldt, 2005)
2. Design automation using CBR (Cederfeldt, 2004)
3. Automated cost estimation in conceptual design (Sandberg et al., 2005)
4. Design tool for distortion assessment (Sandberg et al., 2005)
5. Combining KBE and CBR for design and manufacturability analysis iteration (Sandberg and Marefat, 2006)
6. KBE support for prefabricated timber housing (Sandberg et al., 2008)
7. Automated design of rotary draw bending tools (Johansson and Sunnersjö, 2006)
8. Automated design of toolsets for rotary draw bending of aluminum tubes (Johansson, 2007)
9. Manufacturability analysis integrating KBE, CAD and FEM (Johansson, 2008)
10. Automated idealization of CAD models for finite element analysis (Stolt, 2005)
11. PM Wizard (Stolt, 2008)
12. Knowledge-enabled pre-processing (Boart et al., 2006)
13. Automatic vehicle packaging (AVP) (Fuxin, 2005)
14. Automated optimization for CAE-driven product development (Merkel and Schumacher, 2003)
15. DART (Pinfold and Chapman, 1999)
16. Schemebuilder (Counsell et al., 1999)
17. ICAD Multi Model Generator (MMG) (La Rocca et al., 2002)
18. Feature-based KBE for base engine design (Weilguny and Gerhard, 2009)
19. Knowledge-based system for cost modeling of aircraft gas turbines (Tamminenia et al., 2009)
20. FlexSim (Raffaelia and Germania, 2008)
21. Automated design of electric discharge machining electrodes (Lee and Li, 2009)
22. Automatic layout design of plastic injection mould cooling system (Li et al., 2005)
23. SEDI (Strinning, 1995)
24. Automated design of a press brake (Colombo et al., 2005)

1. CoRPP *Motivation:* Time-consuming and repetitive tasks related to design for cost *Purpose:* To demonstrate a framework of integrated applications for automation of design and cost analysis *Identification of the particular application:* General problem found in a specific company *Methodical approach:* Software development *Architectural considerations:* System architecture *Process improvement/knowledge management:* Process improvement *KM related considerations:* Knowledge modeling	**2. Design automation using CBR** *Motivation:* Time-consuming tasks related to design of variants *Purpose:* To demonstrate a design automation system utilizing case-based reasoning (CBR) *Identification of the particular application:* General problem found in a specific company *Methodical approach:* Software development *Architectural considerations:* System architecture *Process improvement/knowledge management:* Process improvement *KM related considerations:* N/A
3. Automated cost estimation in conceptual design *Motivation:* Repetitive tasks related to cost estimation are found suitable for KBE *Purpose:* Embed a framework for life-cycle cost estimation *Identification of the particular application:* General problem found in a specific company *Methodical approach:* N/A	**4. A design tool for distortion assessment** *Motivation:* Repetitive and routine tasks related to manufacturability are found suitable for KBE *Purpose:* To merge KBE and non-linear finite element analysis *Identification of the particular application:* General problem found in a specific company *Methodical approach:* N/A

Architectural considerations: N/A *Process improvement/knowledge management:* Process improvement *KM related considerations:* Briefly described knowledge acquisition and formalization	*Architectural considerations:* System architecture *Process improvement/knowledge management:* Process improvement *KM related considerations:* N/A
5. Combining KBE and CBR for design and manufacturability analysis iteration *Motivation:* Automation of time-demanding and repetitive tasks related to iteration between design and manufacturability analysis *Purpose:* To combine KBE and CBR to perform "what-if" studies *Identification of the particular application:* General problem found in a specific company *Methodical approach:* N/A *Architectural considerations:* System architecture *Process improvement/knowledge management:* Process improvement *KM related considerations:* N/A	**6. KBE support for pre-fabricated timber housing** *Motivation:* Automation of time-demanding tasks. Make engineering knowledge available. *Purpose:* Apply KBE in the construction business *Identification of the particular application:* General problem found in a specific company *Methodical approach:* Software development *Architectural considerations:* Consideration of business objectives, processes and IT with focus on IT *Process improvement/knowledge management:* Process improvement and knowledge management *KM related considerations:* Knowledge acquisition and formalization. Addresses "knowledge bottlenecks".
7. Automated design of rotary draw bending tools *Motivation:* Routine tasks are identified in the design of rotary draw bending tools that can be automated.	**8. Automated design of toolsets for rotary draw bending of aluminum tubes** *Motivation:* Automation of repetitive and routine tasks to speed up design of fixtures for production preparation.

7. Automated design of rotary draw bending tools (continued) *Purpose:* Control of generic CAD models with heuristic and algorithmic knowledge. *Identification of the particular application:* General problem found in a specific company *Methodical approach:* N/A *Architectural considerations:* N/A *Process improvement/knowledge management:* Process improvement *KM related considerations:* N/A	**8. Automated design of toolsets for rotary draw bending of aluminium tubes (continued)** *Purpose:* To exemplify how a design automation system is developed based on knowledge pieces implemented in auxiliary software. *Identification of the particular application:* General problem found in a specific company *Methodical approach:* N/A *Architectural considerations:* System architecture *Process improvement/knowledge management:* Process improvement *KM related considerations:* Knowledge modeling. Brief discussion of knowledge types and validity.
9. Manufacturability analysis integrating KBE, CAD and FEM *Motivation:* Manual processes for setting up FE-analysis for manufacturability is formalized and analyzed. *Purpose:* Use of an inference engine to handle knowledge objects which connect to auxiliary software applications. *Identification of the particular application:* General problem found in a specific company *Methodical approach:* N/A	**10. Automated idealization of CAD models for finite element analysis** *Motivation:* Minimize time spent on manual idealization of CAD geometry for FE-analysis through automation *Purpose:* Implement the system as a CAD-integrated application and demonstrate a method for creating mid-surfaces *Identification of the particular application:* General problem found in a specific company *Methodical approach:* N/A

9. Manufacturability analysis integrating KBE, CAD and FEM (continued) *Architectural considerations:* System architecture *Process improvement/knowledge management:* Process improvement *KM related considerations:* Knowledge modeling	**10. Automated idealization of CAD models for finite element analysis (continued)** *Architectural considerations:* N/A *Process improvement/knowledge management:* Process improvement *KM related considerations:* N/A
11. PM Wizard *Motivation:* Enable the design of parts for powder metallurgy (PM) through automated analysis, recommendations and tool design *Purpose:* Implement a set of rules for design for PM integrated in a CAD environment *Identification of the particular application:* General problem found in a specific company *Methodical approach:* N/A *Architectural considerations:* System architecture *Process improvement/knowledge management:* Process improvement and knowledge management *KM related considerations:* An interface integrated in the CAD system for continuous knowledge capture related to geometries is proposed.	**12. Knowledge-enabled pre-processing** *Motivation:* Automate pre-processing for FE-analysis. *Purpose:* Implement a CAD integrated pre-processor for FE-analysis *Identification of the particular application:* General problem found in a specific company *Methodical approach:* N/A *Architectural considerations:* N/A *Process improvement/knowledge management:* Process improvement *KM related considerations:* N/A

13. Automatic Vehicle Packaging, AVP

Motivation:
Automation of configuration and virtual assembly of digital mock-ups.

Purpose:
Implement a framework for geometry management together with a solution to integrate many sources of geometry-related information in the process of creating a digital mock-up.

Identification of the particular application:
General problem found in a specific company

Methodical approach:
N/A

Architectural considerations:
Considerations of the enterprise architecture from business strategies to information integration are provided (with heavy focus on IT)

Process improvement/knowledge management:
Process improvement

KM related considerations:
A discussion regarding the cost-benefit of increased knowledge management (which is equated with increased documentation) is provided.

14. Automated optimization for CAE-driven product development

Motivation:
Integrated CAD/CAE optimization loop to automatically design-analyze-redesign variants of a component.

Purpose:
Demonstrate a parameterized component modeled in both CAD and CAE with a rule set governing design and redesign based on CAE results.

Identification of the particular application:
General problem found in a specific company

Methodical approach:
N/A

Architectural considerations:
N/A

Process improvement/knowledge management:
Process improvement

KM related considerations:
N/A

15. DART

Motivation:
Automate meshing of an automotive body-in-white

Purpose:
Unify product and process models in the areas of CAD and FEA into a single model containing engineering intent for faster assessment of novel designs.

16. Schemebuilder

Motivation:
Need to enable a knowledge-based cross-domain computer support tool for fast design of mechatronic systems.

Purpose:
To integrate a generic model of mechatronic systems containing an ontology and abilities to store knowledge and automate design and evaluation of mechatronic systems design.

15. DART (continued) *Identification of the particular application:* General automotive body-in-white problem *Methodical approach:* RAD (software development) *Architectural considerations:* System architecture *Process improvement/knowledge management:* Process improvement and knowledge management *KM related considerations:* Knowledge acquisition, structuring and maintenance	**16. Schemebuilder (continued)** *Identification of the particular application:* General problem in design of mechatronic systems *Methodical approach:* N/A *Architectural considerations:* System architecture *Process improvement/knowledge management:* Process improvement and knowledge management *KM related considerations:* Knowledge modeling, knowledge capture and reuse
17. ICAD Multi Model Generator (MMG) *Motivation:* Need to reduce time spent on variant design as well as time spent on creation, conversion and preparation of design models to various analysis models *Purpose:* Demonstrate a single "master model" (implemented in ICAD) that is rules based and from which all other models for different kinds of analysis are derived. *Identification of the particular application:* General problem demonstrated in the case of aviation industry *Methodical approach:* N/A *Architectural considerations:* Interfaces towards other IT systems are a core consideration but implemented without architectural considerations.	**18. Feature-based KBE for base engine design** *Motivation:* The high-level strategy for shorter lead times and increased quality is concretized in the case of an automotive OEMs base engine to be in automating the design of bores. *Purpose:* To demonstrate a KBE for automating design using geometric feature technology as knowledge carriers between different systems *Identification of the particular application:* General problem identified in a particular automotive company *Methodical approach:* N/A *Architectural considerations:* Implemented with architectural considerations related to both processes (information needs and processing) and IT (system interfaces).

17. ICAD Multi Model Generator (MMG) (continued) *Process improvement/knowledge management:* Process improvement *KM related considerations:* N/A	**18. Feature-based KBE for base engine design (continued)** *Process improvement/knowledge management:* Process improvement and knowledge management *KM related considerations:* Addresses in a structured way the change management issues such as motivation and introduction of a KBE application into a process.
19. Knowledge-based system for cost modeling of aircraft gas turbines *Motivation:* Need to provide more accurate cost estimations quicker and earlier in the design process. *Purpose:* To demonstrate a methodology for representing cost information coupled to product definition. *Identification of the particular application:* General problem demonstrated in the case of aviation industry. *Methodical approach:* N/A *Architectural considerations:* N/A *Process improvement/knowledge management:* Process improvement *KM related considerations:* Considerations mainly related to knowledge capture and knowledge modeling.	**20. FlexSim – knowledge-based system for design of flexible parts** *Motivation:* Design of flexible parts such as hoses or cables is subject to a time-consuming and iterative process while being rather rule-driven and easily automated *Purpose:* To demonstrate a framework of methods and systems to support automatic design of flexible parts. *Identification of the particular application:* General problem in the design of systems requiring flexible parts such as hose and cables. *Methodical approach:* Alternative solution methods for each part of the system are considered but no methodology for knowledge-based systems is used. *Architectural considerations:* N/A *Process improvement/knowledge management:* Process improvement *KM related considerations:* Some issues related to different types of knowledge are considered.

21. Automated design of electric discharge machining (EDM) electrodes

Motivation:
Design of electrodes for the EDM process used when making injection molds for molding fine plastic components is time-consuming while rather rule-driven and easily automated.

Purpose:
To implement a tool using a specific method for splitting the electrode geometry in a CAD system.

Identification of the particular application:
General problem in the design of electrodes for the EDM process.

Methodical approach:
N/A

Architectural considerations:
N/A

Process improvement/knowledge management:
Process improvement

KM related considerations:
N/A

22. Automatic layout design of plastic injection mould cooling system

Motivation:
Preliminary or layout design of the cooling system for plastic injection moulds can be automated to save time and consider several layouts before going into detail design.

Purpose:
To implement a tool using a specific method for creating cooling features based on the plastic part features.

Identification of the particular application:
General problem in the design of injection moulds.

Methodical approach:
N/A

Architectural considerations:
N/A

Process improvement/knowledge management:
Process improvement

KM related considerations:
N/A

23. SEDI (Semi-Empirical Design of Impellers)	24. Automated design of hydraulic press brakes
Motivation: Need to reduce the costs and lead times related to iteration between testing and design. *Purpose:* To implement a tool for automatic design and simulation to support the trial-and-error design process of hydraulic pump blades. *Identification of the particular application:* Solution based on an internal assessment of process issues and implemented by expert designers. *Methodical approach:* N/A *Architectural considerations:* N/A *Process improvement/knowledge management:* Process improvement *KM related considerations:* N/A	*Motivation:* Design of press brakes for sheet metal bending is time consuming while being rather rule-driven and easily automated *Purpose:* To implement a tool for automatic design of press brake families. *Identification of the particular application:* N/A *Methodical approach:* N/A *Architectural considerations:* N/A *Process improvement/knowledge management:* Process improvement *KM related considerations:* N/A

Paper A

Towards Integration of KBE and PLM

Ćatić, A. and Malmqvist J. (2007)

Proceedings of International Conference on Engineering Design – ICED07, Paper no. 540, 28-31 August, Paris France

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED07
28 - 31 AUGUST 2007, CITÉ DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

TOWARDS INTEGRATION OF KBE AND PLM

Amer Ćatić[1], Johan Malmqvist[1]
[1]Chalmers University of Technology, Göteborg, Sweden

ABSTRACT
In this paper the issue of integrating knowledge based engineering (KBE) and product lifecycle management (PLM) is addressed at an architectural level. State of the practice and state of the art KBE applications in the literature and in industrial use constituted the empirical base for a categorization of such applications. Two categorizations are presented; one where applications are viewed from the perspective of desired result and one which relates the KBE application to the task it performs and the tool it performs it with. A service oriented PLM architecture has been found to be promising for integrating KBE and PLM. PLM services, whose main role is to retrieve and store data needed or generated by the KBE application, constitute the integration pattern. The KBE applications, which apply PLM services, are in their turn offered as services provided through the PLM environment. Based on these KBE services, a concept called knowledge modules is introduced. The aim of knowledge modules is to map and integrate KBE applications to support or automate engineering activities which today are performed as services between engineering departments such as e.g. complete design verifications or complete configurations.

Keywords: knowledge based engineering, KBE, product lifecycle management, PLM

1 INTRODUCTION

When computers were introduced as tools in engineering work the idea of using them to capture and reuse engineering knowledge arose. The first systems of this kind appeared in the mid 1970's, so-called expert systems used for rationalizing different activities such as problem solving, calculations, simulation, configuration, design and so on [1]. Later on the use of these kinds of systems in engineering was labelled Knowledge Based Engineering (KBE) referring to the fact that some sort of explicit engineering knowledge is embedded in them. Since then, many examples illustrating the potential of this technology have been demonstrated in the literature [1-11].
Applications with embedded knowledge in use in the industry today are mostly developed internally. The applications vary in size and number. Some advanced solutions are integrated in computer aided design/engineering (CAD/CAE) applications [2,4,11] but most of the applications are small and performing partial tasks such as a spreadsheet performing standard calculations as part of a more extensive engineering task. The applications are also for the most part very loosely, if at all, integrated with product data management (PDM) and computer aided design/manufacturing/engineering/other (CAD/CAM/CAE/CAx) systems [12]. A typical situation is depicted in Figure 1 where the KBE applications are isolated islands and with humans as the only interface towards other systems.
The situation depicted in Figure 1 clearly illustrates the issue of a need for better management of the knowledge in product development. This is realized by an implementation of a strategy for application

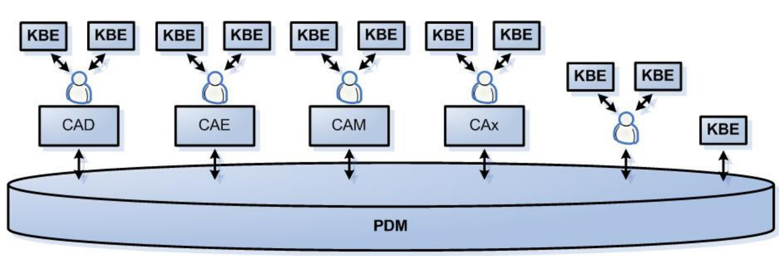


Figure 1 – Current situation of disintegrated KBE applications

of KBE in product development which, in this case, translates to applying a structured approach of integrating KBE with product lifecycle management (PLM).

The development of CAD systems has brought about a tool for more efficient capture and reuse of product data. The introduction of PDM brought about a better way of managing product data. It made it possible to manage the complexity of ensuring a consistent product description at different abstraction levels through versioning, storing, managing access to and structuring the product data created by CAD. Now there is a need to manage the knowledge, from which the product data has resulted, in the same way. However, there is an essential difference between integration of CAD and PDM and integration of KBE and PDM. In the case of CAD integration the management, in terms of e.g. change management or access levels, is facilitated by the fact that the data consists of files and metadata as a package which is easy to manage. In the case of KBE the data to be managed is more fine-grained, e.g. design rules, which makes it harder to manage since the data management needs to be done at such a low level, e.g. versioning individual design rules. Before this issue is addressed the issue of finding a structured approach to transforming the state of KBE applications from being isolated islands, as depicted in Figure 1, towards being an integral part of the PLM environment needs to be addressed. This strategy for the integration of KBE with the PLM environment should, among other things, contain a definition and a categorization of KBE along with an architecture for integration of existing and new KBE applications with the product lifecycle systems. The aim of this paper is to propose such a strategy along with a system architecture that will support its realization.

The paper is structured as follows. The first two sections describe the research approach undertaken along with a summary of state of the practice and state of the art of KBE applications. After that two ways of categorizing KBE applications are presented and finally a proposed architecture for structuring and bringing KBE towards integration with the PLM environment is described with the aim of transforming KBE from being isolated islands to a part of the PLM environment.

2 RESEARCH APPROACH

This study was started with performance of a literature study in the fields of expert systems[13,14], design automation[3-7,9], configuration[2,9], computer aided engineering[4,5,8,10,12] and product lifecycle management[15-18]. This was done in order to find applications implementing KBE; their functionality, implementation and development. A number of applications were found and this constituted the empirical base for a categorization of KBE.

The categorization was finalized and verified by interviewing engineers who have developed and are developing applications which implement KBE to support product development either in tasks relating to CAD or CAE. Along with this the proposed architecture for integration of KBE and PLM using the newly proposed concept of 'knowledge modules', presented in Section 5.4, was verified in the interviews. The interviewed engineers are active within the Volvo Group.

3 RELATED WORK

The term Knowledge Based Engineering has been used in different contexts ever since its introduction. There is still not a universally accepted definition of exactly what is embraced by the term. There are some different definitions. MOKA [1] defines KBE as: *"The use of advanced software techniques to capture and re-use product and process knowledge in an integrated way"*. CommonKADS [19] does not have a definition of KBE but they define the term "knowledge systems" as a gathering term for expert systems, knowledge intensive information systems and knowledge based systems. Another term called "knowledge engineering" is defined as evolved from the art of building those systems.. Pinfold and Chapman [10] define KBE as a derivative of CAD where, besides geometry, also the design intent is captured and therefore KBE is defined as a framework for capturing and defining the process of design creation. Penoyer et. al. [12] define KBE as computer systems used for engineering, focused on a representation and application of knowledge to specific problem cases, has deep penetration into the problem domain and reasons through the problem solving process using rules of logic rather than mathematical models. Poenisch and Clark (2006) [20] confirm that no definition of KBE has found a general acceptance yet but they believe that an essential ingredient of KBE is a software application that processes some kind of engineering knowledge. In this paper KBE is, in a broad sense, considered to include all kinds of applications whose intent is to capture and reuse engineering knowledge, with the term application not being delimited to any particular type of computer software, e.g. ICAD [21] or CAD integrated KBE module such as Catia V5 Knowledgeware [22], UGS NX Knowledge Fusion [23] and Pro/Engineer's Pro/Program [24].

3.1 KBE Tools

The field of design automation is a typical example of KBE. Applications present in the design automation literature describe computer systems which apply engineering knowledge to automate design tasks in order to save time, relieve engineers from tedious tasks and assure that every instance of the designed component is designed using the same rules, thus ensuring a certain level of quality and standardization.

What is common for all examples of design automation applications found [2-7,10] is that they are implemented on mature components for which all of both product and process knowledge is known. This has also been identified by [7] as one of the basic prerequisites for a successful design automation system. The main focus of these applications is the time saving aspect [19] but they also consider other aspects such as different DFX techniques for increased producibility, higher quality or provide information for better decision making such as e.g. cost estimation for different variants [6]. Many of the mentioned applications are mostly concerned with activities related to the synthesis steps of product development. There are applications which demonstrate examples of automation of both synthesis and analysis activities [4,5,8,10].

Most of the applications above will produce predictable results due to the fact that they deal with a low level of uncertainty (either there is one correct outcome or the maturity of the component is high with a high number of known defining rules). Systems which deal with more uncertainty and where the outcome is not as easy to predict are those applying techniques such as case-based reasoning (CBR)[13,14], neural networks (NN) [13,14] or optimization loops [9]. The purpose of these systems is to navigate through a larger space of solutions. The biggest advantage of such systems is that they have the ability to produce results which human engineers might not have found themselves, at least not in the same amount of time [4,9].

3.2 KBE Methods

The proposed methods for developing KBE applications found in the studied literature are the Methodology for Knowledge based engineering Applications called MOKA [1], the methodology for support of knowledge engineering called CommonKADS (Knowledge Acquisition and Documentation Structuring) [19] and Cederfeldt's methodology [7] for planning design automation systems.

MOKA is a result of research done within the European ESPRIT IT Programme by a consortium whose members consist of several European universities and companies from the automotive and aerospace industries. MOKA proposes a structured way of acquiring, structuring and representing knowledge in order for the knowledge to be used in a KBE application.

CommonKADS was developed at the University of Amsterdam also within the European ESPRIT IT Programme as a respond to the need for a standard for knowledge based systems. CommonKADS has a direction towards knowledge in general implying that they could handle not only engineering knowledge but also other types of knowledge such as diagnosis, scheduling of activities and so on. Both of these methodologies propose a structured way of acquiring, structuring and representing knowledge; engineering or other. Both of them also have a clear focus and intention on implementation of knowledge in some kind of stand alone software.

Cederfeldt's methodology [7] gives a basis for primarily the planning of design automation systems through decision criteria regarding identification of possible components for design automation. The methodology also provides support in whether the knowledge should be stored inside or outside the CAD model, the mapping of design rules, how the CAD model should be defined and so on.

3.3 Product Lifecycle Management

The area of product lifecycle management (PLM) is a vast area embracing many disciplines. CIMdata [15] provides the following definition of PLM:

Product Lifecycle Management (PLM) is a strategic business approach that applies a consistent set of business solutions in support of the collaborative creation, management, dissemination, and use of product definition information across the extended enterprise from concept to end of life – integrating people, processes, and information.

An even broader definition of PLM is provided by John Stark [16]:

PLM is the business activity of managing a company's products all the way across their lifecycles, from the very first idea for a product all the way through until it is retired and disposed of, in the most effective way.

In this paper a narrow definition of PLM relating to tools used and processes involved will be considered for KBE integration. The first delimitation will be a focus on so called product lifecycle systems [12] which usually are referred to as PDM and CAx. PDM is viewed as a concept consisting of one or a system of several applications which will, besides product defining data such as geometry, assembly relations, functional relations and requirements also entail analysis and test results, manufacturing data, configurations and so on. CAx stands for the computer based tools the engineers use to author product and process data, the x is thus replaced by design, manufacturing, engineering, process planning, requirements management and so on where each of CAx-s consist of either one or a system of applications. A schematic view of the considered situation is depicted in Figure 2.

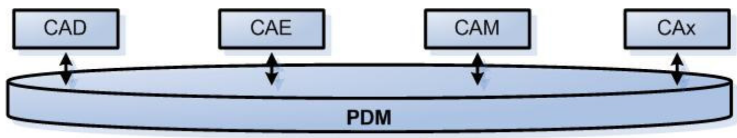


Figure 2 – Product lifecycle systems

Definitions of PLM, such as the one provided by CIMdata, cover all the stages and processes that take place during a products lifecycle, from idea to disposal. In the same manner as for the tools used this paper will consider the earlier stages of a product lifecycle as depicted in Figure 3. These stages provide an overview of how the product is viewed during its lifecycle.

Figure 3 – Product lifecycle stages [25]

3.4 Conclusion

As can be seen there are many examples which demonstrate the capabilities of KBE applications to improve the outcome of engineering work and simultaneously rationalize it. The described methods provide a basis for creating new KBE applications from as well the technical software as knowledge capture and representation point of view. The issue of integrating KBE with PLM has been addressed at a general level [12] but there is so far no proposed way of going from the current situation towards a more structured way of integration. As was mentioned in Section 1 there are two issues relating to this integration. The first one is the fine-grained nature of data in KBE applications which makes it more difficult to manage. The second is the lack of a structured approach to and view of KBE applications. The addressing of the first issue, which is at a more detailed level, needs an addressing of the second issue which is at a more general level. The structured approach needs to provide support and a holistic view of KBE from a product development perspective. This also implies that the implementation of KBE will differ at different sites depending on the product development process it supports and the proposed approach needs to provide the flexibility needed to meet this requirement. The need for flexibility reflects on the proposed system architecture, which addresses the first issue.

4 CATEGORIZATIONS OF KBE

In this section two ways of categorizing KBE are presented. The first is based on whether the focus of the KBE application primarily is automation or increased quality of the solution. The second categorization relates the engineering task, which the KBE application rationalizes, to the tool in which the KBE application is realized. The common requirement for both categorizations is that the applications they categorize capture and reuse engineering knowledge.

4.1 Result oriented categorization of KBE

As the description implies this categorization of KBE applications is based on what is the desired result from the application. Some KBE applications focus on improving quality or performance of the solution while other applications primarily focus on saving time thus reducing cost. Both do it by applying knowledge. This discussion is delimited to applications which only handle explicit knowledge. The categorization is depicted in Figure 4.

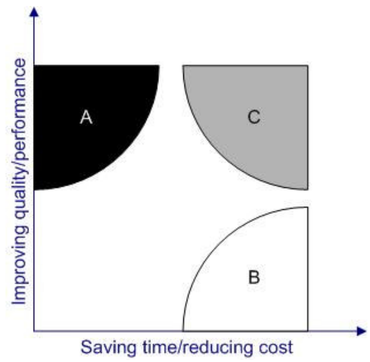


Figure 4 – Categorization of KBE applications according to the expected result

A – Examples of applications in this field are books containing guidelines, reports containing lessons learned from past projects, software applications which have a guiding role e.g. for material selection or CAD integrated warnings for e.g. sharp edges and databases containing documentation of e.g. QFD, FMEA. Examples found in this field are [7,22,26-28]
B – The aim here is to automate repetitive engineering tasks. Examples are applications for standard calculations such as bearing or screw joint dimensioning, quality statistics or parameterized components for which the geometry defining rules are well known [2-7,10].
C – Applications in this field are those who deploy some kind of optimization loops e.g. [4,9].
The fourth corner of the graph is not discussed since there is no point in having KBE applications which neither improve the solution nor save time/cost.

4.2 Task oriented categorization of KBE

The activities performed during development of a product can generally be described by the process depicted in Figure 5. In this categorization focus will be on the first three activities.

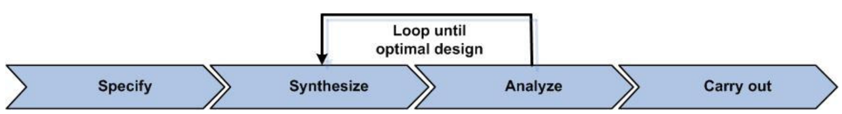


Figure 5 – Product development activities [29]

During these activities engineers author product data. The two kinds of knowledge involved in this work are knowledge about the product and knowledge about the processes used in the development of the product [1]. For the performance of these activities engineers use different tools. Originally the tools used were drawing boards together with equipment needed to test, measure, calculate, produce prototypes and so on. It can be argued that the first approach to capture and reuse engineering knowledge were media containing design guidelines, e.g. sketches from the 15th century depicting exemplifying solutions for different problems. Later on, in the trails of the rising industrialism, handbooks containing design and calculation guidelines along with standards became common applications for engineering knowledge management. The use of books as KBE applications reflects the engineering tools used at the time. All engineering activities were performed manually and thus the KBE applications were also manual in the form of handbooks. Today engineers use a wider range of tools as they create product data; computer aided (CAx) systems with different kinds of dedicated software. The existing KBE applications in industrial use are realized in the same software tools.

These ideas lead to the viewing of KBE and KBE applications, in general, to be applications whose main purpose is to capture and reuse engineering knowledge, as was stated in Section 3. This implies that the nature of the KBE application has to be put in relation to the engineering tool by which it is realized. The requirement on KBE to entail a particular kind of computer software, as many of the found definitions require, results in KBE only embracing applications related to engineering software tools. This neglects the fact that "an engineer's notebook" still is a tool widely used by many engineers [30]. If the requirement is posed in terms of "advanced software techniques" [1], KBE applications realized in e.g. spreadsheets performing simple calculations and saving time would be scoped out.

If taken to the other extreme this discussion also implies that engineers themselves can be viewed as KBE applications implemented in the engineering tool of the human brain. From this perspective a mapping showing the distribution of the total knowledge required in engineering work can be done. From this the following three categories of tasks can be deducted:

- **Creative tasks** which require humans to use their experience, creativity and imagination to produce product data [31]. The "KBE applications" needed for the execution of these tasks are humans executing implicit (non-expressible) knowledge. Both product and process knowledge are possessed by the engineer.
- **Semi-standardized** tasks which require humans to use standard formulas or rules to produce product data. The KBE application for this type of task usually relies on a human for execution and a media, such as a book or a database, for storing the explicit knowledge. This can be viewed as a semi-automatic KBE application where one of either process or product knowledge are contained in the KBE application and the other is possessed by the engineer.
- **Standardized tasks** which can be executed without human intervention. These KBE applications rely on computers for execution and a digital knowledge base whose form is such that it is suitable for computer execution. This can be viewed as an automatic KBE application which contains both product and process knowledge.

There are two reasons for why the semi-standardized tasks might require a semi-automatic KBE application and not an automatic one. The first reason is that the knowledge about the task execution (process knowledge) might be of implicit form and thus can not be expressed in a suitable way for computer execution. The second reason is that the product knowledge might have uncertainties which require further clarification upon execution which can only be done by an engineer. Cederfeldt [8] has referred to this as a relation between how many design rules that are known and how many design rules there are. Here "design rules" refer to as well product as process knowledge. The closer this relation is to unity the higher is the potential of a design automation application for that particular design task. Adding the dimension of type of task to product development activities gives the categorization in Table 1. For every category there are applications listed. These are only examples which illustrate the different KBE applications used in the different cases.

Table 1 – Categorization of KBE applications with examples

	Specification	**Synthesis**	**Analysis**	
Creative tasks	Designer/Marketer	Designer + TRIZ [27]	Analyst	**Human**
Semi-standard tasks	Requirements database	Design guidelines [22][26][28]	Calculation guidelines	**Semi-automatic KBE application**
Standard tasks	Feature packages Options	Parameterized solid + Rule Base [3][5][7]	Automatic simulation program[5][11]	**Automatic KBE application**

5 KBE AS SERVICES IN PLM

In this section an architecture for integrating KBE and PLM is proposed. The proposed solution is delimited to automatic and semi-automatic KBE-applications according to the categorization presented in section 4. This delimitation is justified by the fact that the general strive in both academia and industry is to make the capture and reuse of knowledge as automatic and digital as possible, as can be observed among the application examples referred to in Section 3 about related work. In addition to this a concept called knowledge modules is introduced.

5.1 Service oriented PLM

The application of a service oriented architecture and mindset in PLM has come up as a trend in recent years which has been illustrated by recent initiatives [17,18,32,33] which strive towards creating a flexible PLM environment with best of breed PLM tools, both commercial and in-house developed, performing specialized tasks. The general idea is that every database where information is stored offers its information as services. These services are basically retrieval or storage of information with basic PDM functionality such as versioning, identification, effectivity and so on. The information is communicated using a communication standard and the information is modelled using an information model standard [33]. This is done for the sake of setting the ontology in the PLM environment. The idea is to use these basic services to enable higher level information integration and visualization from heterogeneous sources from different engineering domains and from different databases in these domains. These basic services are referred to as PLM services [17,33]. To make sure that all the databases containing different kinds of information can offer this information as services there is a need for some kind of translator, schema mapping or ontology mapping mechanism to ensure that the services provided by the different information sources comply with the overall standard in the PLM environment.

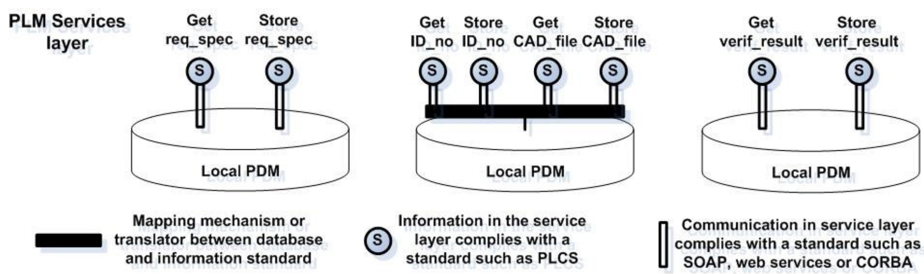


Figure 6 – PLM services

5.2 KBE and PLM services

Based on the above mentioned PLM services performing retrieval and storage of information there is a higher potential for integrating KBE with PLM by implementing PLM services to enable KBE applications to communicate with the PLM environment. This integration has several benefits. When integrated; the KBE applications can be easier created since more of the information needed, such as product models, configurations, analytical results and so on, can be retrieved from other sources and thus do not need to be created locally in the application for its sole purpose. There is a higher possibility of separating the storage of the knowledge from storage of the models executed by the application, e.g. the storage and versioning of a CAD model can be done in a CAD vault while the rule base, which controls the parameters in the CAD model, can be stored elsewhere without bothering the user to store them separately and in different databases. This is done automatically as the application uses different PLM services for the different actions. This example is depicted in Figure 7.

5.3 KBE as services

In the same manner as PLM services KBE applications can be modelled as services in the PLM environment. These services would be performed by automatic KBE applications performing such engineering tasks for which there is a high level of maturity in both product and process knowledge which makes them suitable for automatic KBE applications according to the categorization in section 4. Examples of such automatic KBE applications could be a synthesis of a solid model by execution of a parameterized CAD model along with a rule base or perhaps a standard calculation of e.g. bearing lifetime by execution of governing formulas along with loads and surrounding parameters. KBE services would be given certain inputs by the user and perform a processing of information according to some intelligence either through some standard predefined process, through application of computational techniques such as CBR or rule sets along with an inference engine. The KBE services rely on the PLM services to retrieve and store the correct information in a correct manner as described in the sub-section above.

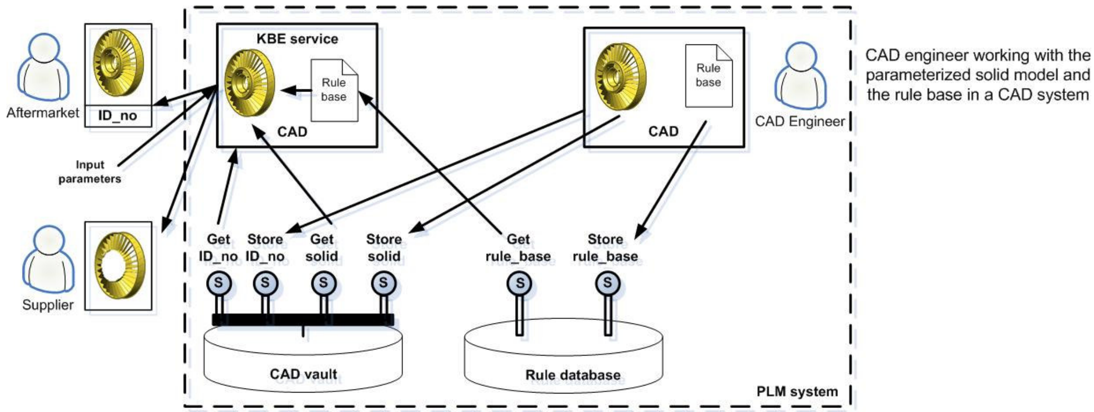


Figure 7 – KBE services

5.4 Knowledge modules

A knowledge module is an integrating entity for KBE applications and it contains both product and process knowledge needed for the performance of an engineering task. The idea of knowledge modules is to integrate KBE applications which perform partial tasks and package them as services across engineering, product lifecycle process or company domains. There is already today a service oriented way of working across engineering and product lifecycle process domains at the observed companies and it is these product information related engineering activities performed as services which should be subjects for knowledge modules. A typical example is an engineer in the design department who makes a request for some kind of analysis (e.g. component strength verification) of a new design. The analysis (verify component strength) is provided as a service by the calculation department. Other examples are simulations and manufacturing adaptations performed as services towards the designer or configurations and design information provision performed by the designer towards sales and aftermarket or a supplier. The idea with the knowledge modules is that these services are provided through the PLM system. The engineering task could be performed by one KBE application or several KBE applications which use each other as services, all depending on the task and the KBE applications themselves, an example is depicted in Figure 10.

The knowledge modules, or rather services which they perform, consist of several activities. The activities can be performed either by humans, by automatic KBE applications or by humans who use semi-automatic KBE applications as support. The knowledge module could be modelled using e.g. the IDEF0 technique [34] depicted in Figure 8.

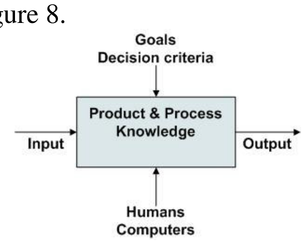


Figure 8 – An IDEF0 model [34] of a knowledge module

The idea of knowledge modules has several objectives. When modelling engineering processes using IDEF0 the performed activities are mapped and their interdependencies are revealed together with their inputs, outputs, mechanisms and controls. An analysis of which of these activities are performed by automatic or semi-automatic KBE applications is done in order to map out known and find unknown applications and finally see where new KBE applications could be, or need to be, developed. This provides a tool for a strategic approach towards applying and developing KBE applications. The proposed process for performing this mapping and analysis is depicted in Figure 9.

The use of IDEF0 to map out the engineering process is just used to exemplify. Any other tool for modelling processes could be used depending on the engineering process to be mapped. The second step in the flowchart, the analysis of the KBE environment could be done by e.g. interviewing the engineers or by performing a survey or a questionnaire. A method and a tool for this kind of analysis is currently being investigated by the authors. In the final step where an implementation of new KBE applications is done the methodology proposed by Cederfeldt [7] could be applied.

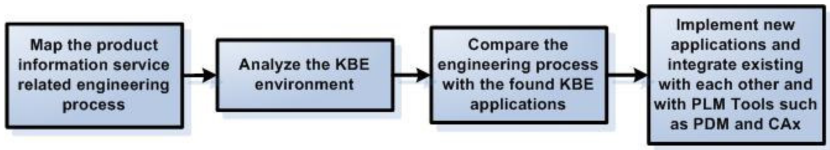


Figure 9 – A strategic approach towards integration of KBE and PLM

The analysis of the engineering processes together with the existing KBE environment also provides information about which KBE applications need to communicate with other PLM engineering tools such as PDM, CAx or requirement management tools. In this paper this translates into which PLM services the KBE applications need. An example of one such mapping is illustrated in Figure 10 where a current process of component strength verification along with the existing KBE environment is illustrated in the upper half. After a discussion of the process and analysis of the KBE environment some possible new KBE applications (offered as KBE services) and PLM services were identified depicted in the lower half of the figure.

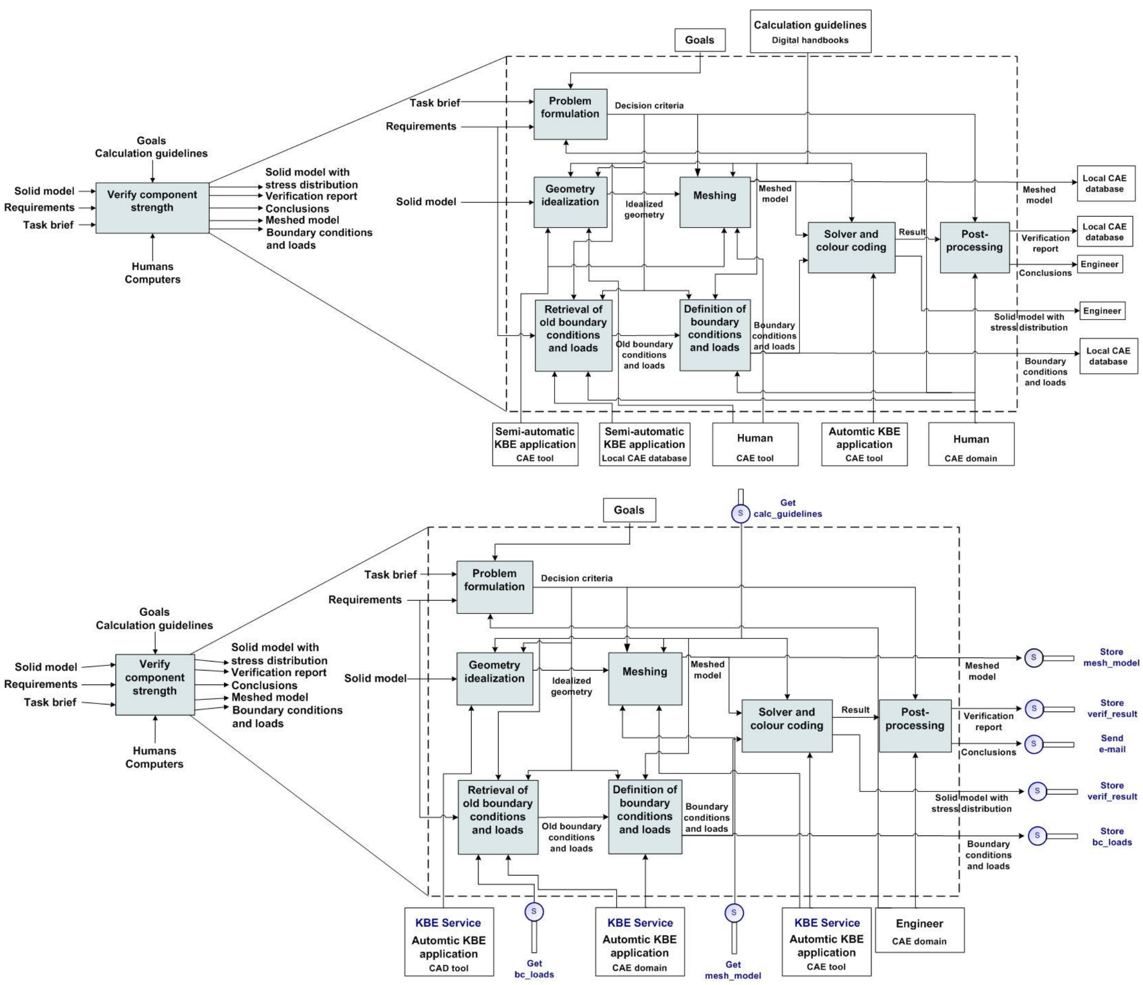


Figure 10 – Upper half: component strength verification process at one observed departments. Lower half: result of process from Figure 9 where some suggestions for new KBE applications were proposed along with application of PLM and KBE services

6 DISCUSSION

The proposed architecture addresses the problem of the lack of a structured approach towards KBE as a supporting tool for product development. There are however many issues related to this to solve. One of them, mentioned in the introduction, is the fine-grained nature of KBE knowledge structure which makes them hard to manage. In the proposed architecture the knowledge is managed in batches in the form of e.g. text or tables containing rule sets meaning that a solid model or a calculation model and the rule sets which belong to them are managed separately. Upon execution the executing application merges the correct model with the correct rule set by some ID number which means that every model needs to have its own rule set even if the rule sets are 90% identical. However this is dependable of the granularity of the PLM services. If for example the database containing design rules for a product platform in which many solid models contain global platform parameters stores the rules for each parameter separately then the PLM services provided by this database need to reflect this granularity. This refinement of PLM service granularity could be a means for managing the fine-grained nature of KBE applications. The need to model the knowledge in the system also becomes more apparent when the level of granularity is reduced opposed to managing rule sets in files. In other words the integration pattern has to be further investigated in order to set the PLM service granularity at a suitable level. This applies especially to PLM services needed for the support of KBE applications. The application of PLM services and KBE services as described in this paper might lead to a need for many translators and small applications for e.g. separation of the rule base and the solid model in CAD. However the use of widely accepted information and communication standards ought to minimize this need. Related to this are also interface issues concerning software packages which are unwilling to share information. This protectionist behaviour is recognized both by [12] and the interviewed engineers who see it as one of the main obstacles for software integration. This behaviour also leads to a conflict between where KBE should be implemented and where it should be used. In this paper it is argued that logically a KBE application should be implemented in the engineering tool used for the task that the KBE application is supposed to perform. The problem with this appears when the results of the KBE application are needed by another KBE application working in some other tool. If the software tools, in which the KBE applications are implemented, do not communicate then these KBE applications remain isolated islands with a need to manually transfer data, if even that is possible. In the worst case, as mentioned by one of the interviewed engineers, there is a need to build and manage two identical models of the same component in the different software tools.

The proposed use of information modelling standards in the service oriented architecture has by some of the interviewed engineers been appointed as an area containing issues related to the fact that some information is lost in the conversion. This is especially true for information related to special functionality in CAx systems. However in the service oriented PLM and KBE the important issue is not which functionality that was applied within the CAx systems to obtain the result but rather that the result can be modelled in a standard format. It is only when information in the result cannot be modelled due to special functionality that this issue appears which to some extent could be resolved by inclusion of metadata.

An important issue which has been discussed in other related literature about KBE and design automation is the need to show the end user of the KBE applications exactly what is going on to avoid the risk of a black-box effect leading to the user being suspicious towards the application due the simple fact that its way of providing the answer is not clear. The KBE services should clearly show which PLM services and other KBE services have been used to retrieve information and how the provided result has been realized.

In this paper, and in the reviewed literature, there is a general aim towards reasoning about the reuse of engineering knowledge from earlier product lifecycle phases in later phases. There is, however, not much discussion about how the knowledge from later stages, such as production and aftermarket, is to be fed back into the earlier stages. This was pointed out by the interviewed engineers as an important issue to consider and further investigate.

7 CONCLUSIONS

From the presented results and ideas in this paper it can be concluded that KBE is still an area without an established definition and lacking methods or approaches for integrating it with product development and PLM. This paper argues that a broad view of KBE is needed in order to entail all applications which capture and reuse knowledge when a structured approach towards integrating KBE with product development and PLM is considered.

It is also argued that KBE applications should be implemented in the software tools which are dedicated to the engineering task which the KBE application is supposed to support or rationalize. This leads to issues concerning that actors within other domains who might have use of the KBE application also need to have the dedicated software. It also leads to issues concerning the need to integrate each software containing KBE applications with the PLM environment separately. These issues are addressed by application of a service oriented architecture where KBE applications use PLM service to retrieve and store data and KBE applications themselves are offered as services towards other actors who don't need to have dedicated software to obtain a result from the KBE application. This service oriented mindset is already applied by the observed and interviewed engineers who perform much of their work as services towards actors in other domains.

It was finally concluded that the service oriented PLM architecture seems promising at an architectural level but there is further investigation needed in the issue of what level of granularity the PLM services should be implemented at to support KBE services.

8 FUTURE WORK

A demonstrator of PLM services, KBE services and knowledge modules will be developed in future work. Other issues that will be more closely investigated are how the management of KBE applications can be done with respect to their fine-grained nature and how the issue of software interfaces is addressed. The issue of applying KBE as a means for feedback of knowledge from later to early phases will also be investigated. Also the proposed process for a more strategic approach towards implementing new KBE applications in product development, see Figure 9, will be further studied.

ACKNOWLEDGEMENTS

This study was funded by the Swedish Governmental Agency for Innovation Systems "VINNOVA" through the Vehicle Research Programme. The authors would like to express their gratitude to Dr. Ola Isaksson at Volvo Aero along with the group of experts from various parts of the Volvo Group who have provided insightful comments about this work. A great deal of gratitude is also accredited to the interviewed engineers for their time and insights in the topic.

REFERENCES

[1] Stokes, M. (2001) *Managing Engineering Knowledge: MOKA: Methodology for Knowledge Based Engineering Applications*, Professional Engineering Publishing Limited, London, UK

[2] Fuxin, F. (2005) *Enabling Preconditions for Geometrical Application in Preconceptual Stages of Engineering Design*, Proceedings of ICED 05, page 79-80, Melbourne, Australia

[3] Colombo, G., Girotti, A., Rovida, E. (2005) *Automatic Design of A Press Brake for Sheet Metal Bending*, Proceedings of ICED 05, page 124-125, Melbourne, Australia

[4] Merkel, M., Schumacher, A. (2003) *An Automated Optimization Process for a CAE Driven Product Development*, Transactions of the ASME, vol. 125, page 694 – 700

[5] Boart, P., Andersson, P., Elfström B-O. (2006) *Knowledge enabled pre-processing for structural analysis*, Proceedings of NordPLM '06, page 89-97 Göteborg, Sweden,

[6] Elgh, F. (2006) *Automated Cost Estimation of Products Variants – A Tool for Enhanced Producibility*, Licentiate thesis, Chalmers University of Technology, Göteborg

[7] Cederfeldt, M. (2005) *Planning Design Automation Systems – Criteria and Guidelines*, Licentiate thesis, Chalmers University of Technology, Göteborg

[8] Bylund, N., Isaksson, O., Kalhori, V., Larsson, T. (2004) *Enhanced Engineering Design Practice using Knowledge Enabled Engineering with Simulation Methods*, Proceedings of DESIGN 2004, page 841-846, Dubrovnik, Croatia

[9] Cederfeldt, M. (2006) *Towards a Strategy for Mapping of Design Problems to Suitable Solutions – A Case of Design Automation Using CBR*, Proceedings of DESIGN 2006, page 471-478, Dubrovnik, Croatia

[10] Pinfold M., Chapman C. (1999) *The Application of Knowledge Based Engineering Techniques to the Finite Element Mesh Generation of an Automotive Body-in-white Structure*, Journal of Engineering Design, Vol. 10, No. 4, page 365-376
[11] Strinning, P. (1995) *A Computerized Trial and Error Technique for Impeller Design*, The ITT Flygt Scientific Magazine, no. 3, pp. 19-27
[12] Penoyer, J.A., Burnett, G., Fawcett D. J., Liou, S.-Y. (2000) *Knowledge Based Product Life Cycle Systems: Principles of Integration of KBE and C3P*, Computer-Aided Design, no. 32, page 311 – 320
[13] Sriram, R. D. (1997) *Intelligent Systems For Engineering*, Springer-Verlag, London, UK
[14] Jackson, P. (1999) *Introduction to Expert Systems*, Addison Wesley Longman Ltd, Harlow, UK
[15] CIMdata, www.cimdata.com, access date: 2006-01-20
[16] Stark, J. (2005) *Product Lifecycle Management: 21st Century Paradigm for Product Realisation*, Springer Verlag Ltd, London, UK
[17] Bergsjö, D., Burr, H., Malvius, D., Mueller, M., Vielhaber, M., Malmqvist, J., (2007) *Product Lifecycle Management for Cross-X Engineering Design*, Proceedings of ICED '07, Paris, France
[18] CIMData (2006) *Service-Oriented Architecture for PLM, an Overview of UGS' SOA Approach*, Ann Arbor, MI, USA
[19] Schreiber, G. et. al. (2000) *Knowledge Engineering and Management: The CommonKADS Methodology*, The MIT Press, Cambridge, Massachusetts, USA
[20] Poensich, V., Clark, A. (2006) *Elements of an Interface Standard for Knowledge Sources in Knowledge-Based Engineering*, ASME Journal of Computing and Information Science in Engineering, vol. 6, no. 1, page 78-83, USA
[21] Knowledge Technologies International (2007) www.ds-kti.com, access date: 2007-01-20
[22] Dassault Systemes (2007) www.3ds.com, access date: 2007-01-20
[23] Unigraphics Solutions (2007) www.ugs.com, access date: 2007-01-20
[24] PTC (2007) www.ptc.com, access date: 2007-01-20
[25] Ulrich, K.T., Eppinger, S.D. (2000) *Product Design and Development*, 2nd edition, McGraw-Hill, New York, USA
[26] Cambridge Engineering Selector – CES (2007) www.grantadesign.com, access date: 2007-01-20
[27] Hipple, J. (2006) *The Use of TRIZ Principle in Consumer Product Design*, TRIZ Journal, June 2006, USA
[28] Thilmany, J. (2004) *Share the Wealth*, Mechanical Engineering, v. 126, no. 12, pp. 32-35
[29] Ullman, D.G. (2003) *The Mechanical Design Process*, 3rd edition, McGraw-Hill, New York, USA
[30] Chakrabarti, A., Kota, S., Rao, N., Chowdary, S. (2005) *A Designer Workbench with Real-time Capture and Reuse of Information on Product Evolution and Rationale*, Proceedings of ICED 05, page 496-497, Melbourne, Australia
[31] Regli, W. C., Hu, X., Atwood, M., Sun, W. (2000) *A Survey of Design Rationale Systems: Approaches, Representation, Capture and Retrieval*, Engineering with Computers, no. 16, page 209-235, Springer-Verlag Ltd, London, UK
[32] Georgiev, I., Ovtcharova, J., Georgiev I. (2007) *Modelling web services for PLM distributed system*, International Journal of Product Lifecycle Management, Vol. 2, No.1 pp. 30 - 49
[33] Object Management Group Inc. (2005) *Product Lifecycle Management Services V2.0*, 250 First Avenue, Needham, MA 02494, USA
[34] FIPSPUB 183 (1993) *Integration Definition For Function Modelling (IDEF0)*, Federal Information Processing Standards, Publication 183, National Institute of Standards and Technology, U.S. Department of Commerce, Springfield, VA, USA

Contact:
Amer Ćatić
Department of Product and Production Development
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Tel: Int +46 31 772 5850
Fax: Int +46 31 772 1375
Email: amer.catic@chalmers.se
URL: www.chalmers.se/ppd/EN/

Paper B

Implementing a Service Oriented PLM architecture Focusing on Support for Engineering Change Management

Bergsjö, D., **Ćatić, A.** and Malmqvist J. (2008) International Journal of Product Lifecycle Management, Vol. 3, No. 4, pp 335-355

Implementing a service-oriented PLM architecture focusing on support for engineering change management

Dag Bergsjö*, Amer Ćatić
and Johan Malmqvist

Department of Product and Production Development,
Chalmers University of Technology,
Göteborg SE-412 96, Sweden
Fax: +46 31 772 1375
E-mail: dagb@chalmers.se
E-mail: amer.catic@chalmers.se
E-mail: joma@chalmers.se
*Corresponding author

Abstract: The aim of the study is to implement a service-oriented architecture (SOA) and evaluate its applicability to an industrial case, using PLM services 2.0. This paper focuses on IT support for engineering change management (ECM). ECM is a cross-functional process including several technology fields and life cycle stages. The product information is accessible from a service layer where it is either used directly by the user or provided as input to knowledge-based engineering (KBE) applications which simulate and analyse the impact of an engineering change. It is concluded that SOA is an efficient architecture that enables integration of KBE applications. PLM services 2.0 is a competent new standard that needs improvements regarding documentation and more detail. In general, it can be concluded that it is necessary to invest more in data management and support capabilities, and that SOA contributes to taking better control of the business logic in comparison with other PLM architectures.

Keywords: PLM; KBE; knowledge-based engineering; SOA; service-oriented architecture; ECM; engineering change management.

Reference to this paper should be made as follows: Bergsjö, D., Ćatić, A. and Malmqvist, J. (2008) 'Implementing a service-oriented PLM architecture focusing on support for engineering change management', *Int. J. Product Lifecycle Management*, Vol. 3, No. 4, pp.335–355.

Biographical notes: Dag Bergsjö is a PhD from the Department of Product and Production Development at Chalmers University, Göteborg, Sweden. His research focuses on PLM architecture. His PhD thesis titled 'Product Lifecycle Management – Architectural and Organizational Perspectives' was published in May 2009. His current research project involves IT support for the systems engineering processes and product platform development for the Swedish automotive and aerospace industry.

Amer Ćatić is a Lic Eng from the Department of Product and Production Development at Chalmers University of Technology, Göteborg, Sweden. His research focuses on PLM with emphasis on capture and reuse of engineering knowledge through knowledge-based engineering. The research aims at optimising the IT-support from a business perspective in product development.

His Licentiate thesis was published in March 2009. His current research project focuses on developing tools which implement lean product development methods for capturing and reusing engineering knowledge. The research is being conducted and implemented in close collaboration with the Swedish automotive and aerospace industry.

Johan Malmqvist is a Chair Professor in Product Development at Chalmers University of Technology, Göteborg, Sweden. His research focuses on development methodologies and IT support for product development (PLM). The results include new design methods and IT tools, as well empirical studies of product development practices. The research is conducted in close collaboration with Swedish industry. Current projects investigate methods and tools for development of product-service systems, PLM architecture and knowledge-based engineering tools.

1 Introduction

The integration of processes and products within a product data management (PDM) system has been difficult to achieve in industry – in particular, integration of different disciplines that have been allowed to evolve separately, such as the software and mechanical engineering disciplines. This integration is essential in order to increase time to market and innovation abilities, particularly in the automotive industry. The general direction of product life cycle management (Ćatić and Andersson, 2008) and product development has implied a single-source strategy for gathering and managing product data in a database system. This is also called the All-in-one integration concept, and focuses on one master PLM system that is directly connected to the user applications. Other types of integration concepts such as peer-to-peer and best-in-class are further elaborated and evaluated in Bergsjö et al. (2006). The single-source strategy has proved hard to realise due to the distributed nature of engineering work and systems, and so far it has been impossible to develop a system which supports this way of working in a complex product development setting.

Burr et al. (2005) show that the integration of PLM systems today is not working properly, resulting in data losses, especially when handovers occur in the development process. Integration between systems can take place at different levels. Burr et al. suggest a Best-in-class integration, where the best systems from each engineering discipline are integrated on a corporate level. Challenges when integrating systems involve defining the master source of the information, the level of integration required, and how processes should be managed. Integration depends also heavily on the legacy of information and the traditions of the company (CIMdata, 2006). There are primarily two approaches to integrating system and information. The first one is system-level integration, where systems communicate with each other through common interfaces and export/import functionalities. The second approach is information-level integration where the systems are integrated on a higher abstraction level, with a common information model (Hallin et al., 2004).

As products become more complex through the addition of new functions which to a high extent rely on mechanical components to interplay with electronic and software components, the prediction of an engineering change's (EC) impact becomes harder and harder to predict and analyse. Simultaneously, the people who develop the final product become geographically and corporally scattered through the involvement of suppliers in the extended enterprise constellation. These trends make it hard to deploy an efficient and comprehensible EC process that involves all interested parties. Performing changes of a product in development is only a small part of the problem. A large part of product development consists of introducing changes to an existing solution, which is the case for many mature products like diesel engines.

Service-oriented architecture (SOA) is an approach to designing software application architectures in order to decouple processes from applications and databases. One part of the SOA approach is the service-oriented integration of applications and databases. The other part is the decoupling of processes from the application and database domain, implying that it is possible to optimise the process and use it to pose requirements on the applications and databases. Specifically, in the scope of this paper is a PLM environment where the PLM processes are not dependent on a rigid server and client architecture. SOA enables integration of systems that are heterogeneous (that have customised information models and processes), and is therefore a possible approach to bridging gaps between different systems under the PLM umbrella. A similar approach has been shown by Abramovici and Bellalouna (2008). The idea is to create services that collect, distribute and even modify information in several databases. These services are then reachable from, for example, the user PLM interface when an activity related to a product function that involves several disciplines has to be carried out. SOA can be enabled by a web-driven architecture through the use of Java services that utilise protocols, such as XML, SOAP and Web Service Description Language (WSDL) for communication between independent tiers (Georgiev et al., 2007).

SOA as an industrial application has been evaluated by Lee et al. (2007). In their view, four aspects are important regarding a SOA: the services, the enterprise service bus, business process management and enterprise portal. The services are defined as providers of reusable business functions in an implementation-independent function that is loosely coupled to other business functions. The service bus is the integration middleware where applications are connected by services. Business process management's main function is to provide integration of scattered systems where SOA would offer a smooth integration. Finally, the enterprise portal is used as the presentation layer where users can take the information provided by the service-oriented PLM system.

In order to standardise the application of web services specifically for PLM systems, Object Management Group (OMG) and Oasis have developed their own set of standards independently. The OMG standard is based on ISO 10303 AP 214 and is recognised under the name PLM Services (Feltes and Lämmer, 2007), where version 2.0 is the latest edition, still under revision as this paper is written (May 2008). The second standard, Oasis PLCS PLM web services definition, is based on AP 239 (Vec-Hub, 2007). This standard is further described in Srinivasan et al. (2008).

So far, it has not been satisfactorily shown how to achieve a proper implementation and evaluation of a SOA-based PLM architecture based on a standard like PLM services 2.0. This paper aims to bridge the gap regarding SOA in general and the possibilities to standardise a SOA-based PLM architecture, and evaluate the standard in order to improve both the standard itself and the practical application of it.

As shown in this section, research often focuses on the IT perspective of integration and architecture. The focus is often on holistic issues such as lower cost, maintenance or availability of information. However, in order to fully demonstrate the benefits of an integrated and possibly automated PLM environment, the engineering perspective must be further investigated. The aim of this research is to bring the research dealing with information and architecture one step closer to the engineering, that is actually show how the work flows can be made more efficient with the help of a SOA. This will be done by utilising an information-intensive engineering scenario that can demonstrate the increase in work output and reduction of time spend on administrative tasks.

The main source for gathering information regarding the processes was interviews as well as workshops and project meetings. The automotive developer showed a great interest in the study, which greatly facilitated its conduct. The developed demonstrator supports management of EC, in terms of both change impact analysis and change process management, by integrating knowledge-based engineering (KBE) applications with the product life cycle management (PLM) environment. The stated research questions for the study were the following:

RQ1: How can efficient ECM be supported by PLM integration?

This question aims to investigate the ability of new emerging technologies within the area of IT support in product development that enables better engineering change management (ECM). This involves process automation within the ECM process and integration of information from a heterogeneous PLM environment, in real time.

RQ2: What are the practical implications from a business perspective of implementing a service-oriented PLM architecture?

This question focuses on the larger architectural picture of integration of applications under the PLM umbrella. In particular, the focus is on emerging standards for service-oriented PLM architecture, and the impacts this will have on the future of business processes such as ECM.

RQ3: How suitable are the available standards in supporting a service-oriented PLM architecture?

This question aims to answer how suitable SOA standards are – whether OMG standard as a neutral standard is going to be supported by commercial vendors, or standards based on commercial packages are preferable.

This paper is structured as follows. Section 2 contains the methodology and the research approach including descriptions of the demonstrators developed. Section 3 is an overview of different PLM architectures and in particular SOA and its applicability to PLM. Section 4 focuses on ECM and the applicability of automated ECM through the addition of KBE applications. Section 5 contains the results from the developed demonstrators. Followed up by a discussion of the results. Finally, the conclusions are presented in Section 6.

2 Methodology

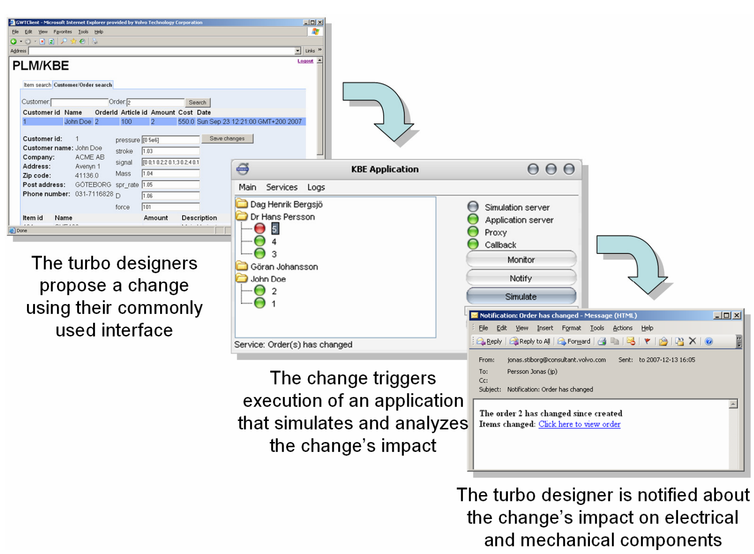
In order to demonstrate the ideas which have been discussed, the work with defining a suitable case for demonstration was initiated. Along with this, an extensive search for different ways of realising a service-oriented PLM architecture was conducted in order to find other implementations and standards which could be applicable. The concept for the demonstrator was discussed and the general idea was that it should demonstrate the implications of service-oriented PLM, from the business, implementer and user points of view. In order to make the demonstrator as realistic as possible, it was decided that an industrial case addressing an existing problem, where this integration could have a substantial engineering benefit, was to be chosen.

2.1 Demonstrator case

It was clear that some kind of multi-domain and multi-organisational issue needed to be addressed in order to demonstrate the idea of integrating heterogeneous environments to support a common process. Further, advanced features such as integration of applications for analysis and evaluation were needed. This led to choosing two similar demonstrator cases with different focuses, one with connection to a real industrial case of a turbo development, and one more advanced case which had the potential to implement less mature technologies in a research context. The chosen case for both demonstrators is ECM in a multi-domain product, thus including involvement from several departments which deal with this issue at an OEM, and also including a supplier. The case for the first demonstrator is about how a change in the turbocharger in a turbo diesel engine affects both mechanical and electrical components in the rest of the engine. The affected components may or may not be in the geometrical vicinity of the turbocharger. The turbocharger is developed and produced by a supplier to the OEM. A caption showing a design engineer's workflow from request for a change in the developed demonstrator is shown in Figure 1.

The second demonstrator case was further expanded to incorporate real-time automated change management of a hydraulic cylinder. For this case, an automated process that included simulation through a Simulink model connected to automatic dimensioning of the hydraulic cylinder in ProEngineer was set up. To add to the complexity, the KBE application controlling the change management and input information was physically located at a different site than the server running the KBE application for simulation and dimensioning of the hydraulic cylinder. The demonstrator cases are further described in Section 6.3.

Figure 1 Demonstrator application (see online version for colours)



2.2 Realisation of a service-oriented PLM architecture

The area of SOA within the computer programming domain is relatively mature, with solutions such as web services based on communication using XML messages according to the SOAP standard. This provides a good basis for making sure that the communication of data is assured. This is, however, not alone sufficient to provide the complete solution for a service-oriented PLM architecture; it provides integration at the system level, but what is needed is integration at the information level, in order to be efficiently connected to the engineering processes. An extensive search for standards and reference cases provided a standard specifically addressing this issue. The standard, called PLM Services 2.0, is provided by the standardisation body OMG. This standard was chosen and implemented in the demonstrator case. It was chosen partly due to the fact that its origin is in the automotive industry, and since then it seemed more mature than the OASIS standard (MacKenzie et al., 2006). Oasis was also evaluated and developed within another branch of the same company (Vec-Hub, 2007). Another contributing factor was that there was an ongoing initiative to look at the corporate PDM and standardise the interfaces according to the PLM services standard. The aim of this paper is not to compare the different PLM SOA standards towards each other, but rather to evaluate the principles behind SOA in a setting close to the practical engineering.

3 From *ad-hoc* to service-oriented architecture

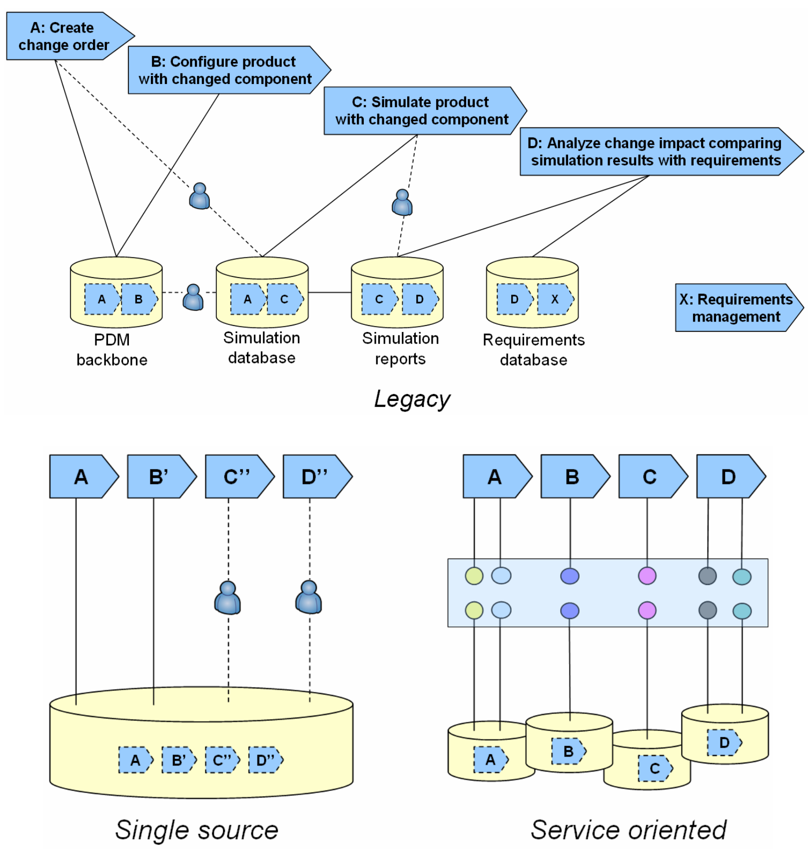
Until recently, the introduction of commercial PDM/PLM software solutions tended to rely on the idea of a single source for all the data. This approach has long roots; when legacy PDM systems were introduced, there was a limited need to integrate different disciplines since for example, mechanical and electrical design could be separated. The idea of a single-source database was further promoted by the fact that the in-house developed legacy systems were, and still are, perceived as a single system. However, the legacy system might have one name but usually consists of many different databases and applications on top of them which process the data. Svensson (2003) points out that the legacy PLM environment tended to be made up of a variety of applications and databases which were implemented every time a business need was recognised. This way of expanding the PLM environment also meant that much of the information was duplicated, and that a lot of time was lost in feeding the same information into different systems. Since the legacy PLM environment usually has a common name, it is natural to think of it as being a system which can be replaced with another system fulfilling the legacy's functions. As the commercial PLM software solutions began to gain competitive functionalities, some companies decided to replace their legacy PLM environments with commercial solutions (Zimmerman, 2008). This shift from legacy to commercial PDM/PLM systems was driven by the increased globalisation, which in some cases meant integrating suppliers in the product development activities, and in other cases meant mergers of companies through acquisitions or partnerships. This is similar to other businesses in the field, for example Dahlén (2006).

Regardless of the cause, increased globalisation required a redefinition of parts of the information models and the process models concerning the corporate traditions and legacy, also known as the business logic. The attempt to replace legacy systems with commercial solutions might also lead to the replacement of those parts of the legacy which were efficient from a company-specific process support point of view, leading to less efficient commercial solutions for those processes (Zimmerman, 2008).

For companies today that need to implement PLM, it is important to realise the value of their processes and business logic on which their entire business and uniqueness rests. Even though it may sound like attempting to cut costs through outsourcing the PLM system by the implementation of commercial, presumably easy to manage software on a globally extended enterprise involving suppliers in all of the product's life cycle stages, it is of critical business value to control and maintain the PLM architecture. This means that companies need to move from a situation of old tools realising a complex business logic towards new tools required for the new business setup, but to keep the essence of the business logic that has proved to be successful in the past.

In Figure 2, a complex legacy environment is exemplified with a business process starting with creation of a change order by the designer in a system. The designer, however, has to notify a simulation engineer who has to initiate a change order in the simulation database that supports the CAE department, which is the reason for the dotted line to the second database with a human interface. In the next step, the simulation step is prepared by configuring the product with the changed product to ensure that the input data for the simulation application are correct, which is done by simulation engineers using the corporate PDM backbone system.

Figure 2 Example of a legacy environment (see online version for colours)



The simulation inputs and outputs such as boundary conditions and load cases are stored in the simulation database, and after the simulation is done the results are summarised in a simulation report, which is reviewed by a reviewer who checks it in the simulation report database. Finally, the designer accesses the report from that database and finds the requirements for the simulated characteristics from a requirements database. In addition to these, there is a human interface between the corporate database and the simulation database, who ensures that the simulation metadata are coupled with the correct article numbers and/or change orders in the corporate PDM system.

Going from the legacy environment to a single source could have the effects depicted in the bottom left of Figure 2, if some events from the legacy environment are preserved. Difficulties might arise if activity B looses efficiency, due to configuration being a very streamlined and complex activity in automotive legacy PDM systems, which is hard to replicate in a new system (Zimmerman, 2008). In the activity C, it is reasonable to assume that the simulation application is not changed, in which case there is a high risk that the format of the simulation results and simulation reports does not correspond to the new single-source system, and thus needs manual processing in between. This fact also slows down activity D since it is likely that this activity's information processing is adjusted to fit with the format coming out of the simulation application. An alternative solution is that activities C and D keep their databases, but then the point of having a single source is lost. Another solution could be to reorganise the complete process and fuse C and D and change the simulation application, but then we would be reorganising our initial process and methods to fit an IT solution.

A service-oriented PLM architecture (bottom right in Figure 2) seems to be a promising solution for creating flexible integration and full process support. Thus, the business processes which the company has built up for decades around its products can be kept. Moreover, the service-oriented PLM architecture enables the process to be further optimised, since the fixed relationship to the legacy databases is not needed anymore. The process is not changed but the human interfaces are gone and the process is more efficient.

The concept of a service-oriented PLM architecture means that the applications and database layer are separated from the business logic and processes, which should not be dependent on the IT tools used.

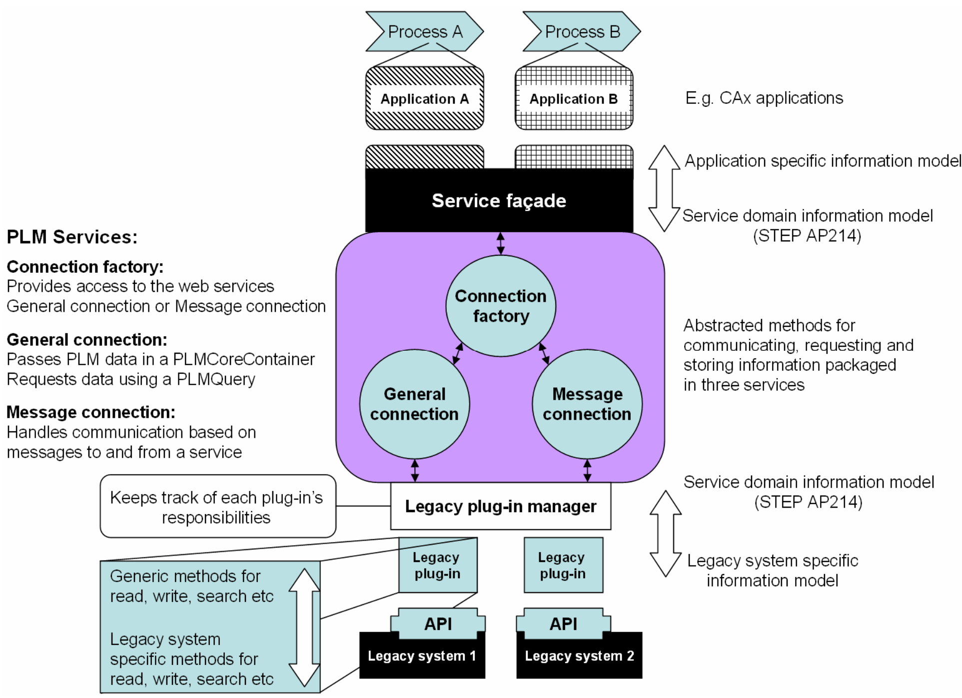
This separation of business logic and processes from the tools is realised by considering applications as providers of information elements, and processes as consumers of these information elements. The layer in between (the middleware) is based on a common contract according to which information elements are expected to be delivered. The contextualisation of these information elements is done in the processes according to the business information model. The simulation application connects to the service layer, and its execution is set to be triggered by a change order initiation inside the corporate PDM system. It then requests a configuration of the variants containing the changed component, and it requests the load cases and boundary conditions belonging to these variants from the simulation database, and the simulation is executed and so on. This provides the processes with the independence from the database layer through loose integration. At the same time, the service-oriented PLM architecture provides the IT governing organisational unit with a possibility to control the IT environment in terms of choosing the best tools for the processes, with the flexibility to change when necessary.

A SOA implies certain requirements on the organisation and management capabilities. Important aspects to consider in order to be successful with a SOA PLM system are: modularity (services built on each other), central coordination (central governance of the service layer), standard communication (facilitating re-use and modularity), use of general modelling constructs (metadata in the service layer that facilitates its management) and minimum process redundancy (services are re-used for same tasks) (Bergsjö et al., 2007). In order to make a SOA-based PLM system work, continuous governance is required. Governance rules and a responsible organisation need to be assigned in order to maintain the processes and IT environment (Bergsjö et al., 2007; Lee et al., 2007).

3.1 Demonstrator architecture

The architecture adopted for the demonstrator is depicted in Figure 3. In the uppermost layers are the applications used in the business process, for example CAx applications. The common attribute possessed by these applications is that they process and create information. They do not store information. In the bottom are the legacy systems which store information, such as PDM systems and other databases.

Figure 3 Demonstrator architecture with PLM Services in the middle (see online version for colours)



Source: Ćatić and Andersson (2008).

The interface between the applications and the PLM services is called the service façade. The service façade keeps track of client sessions so that this does not have to be implemented in each client application, and it also contains a mapping between the information model of the client application and the service domain. On the service side of the mapping, the information model is the STEP AP214 standard, and on the application side it is every particular application's information model. Applications which can read and/or write according to AP214 thus do not need a mapping, which reduces the effort of connecting applications.

For the service domain, the standard PLM Services 2.0 has been used as the enabler for the SOA integration. PLM Services 2.0 is provided by the standardisation body OMG (2006) and has been developed together with representatives from the German automotive industry. It provides the developer with a set of rules and guidelines, a contract, according to which PLM information is communicated, requested and delivered. The PLM Services 2.0 specialty is that its starting points are the common workflows encountered in the PLM area, which should not be confused with processes and workflows embedded in commercial PLM software suites. The workflows are described on a more generic level (since they are not restricted to any particular PDM or CAx software), which means there is flexibility for company specific processes, applications and information.

The standard defines a STEP AP214 compliant data model and all the necessary functionality to realise several use cases. OMG supplies the XML schemes and WSDL files that define PLM Services. The WSDL files specify three web services, Connection Factory, General Connection and Message Connection. Each service is built up of a set of methods specified by the standard. The Connection Factory service handles authentication and creation of sessions and acts as a gateway to the two other services. The General Connection service handles communication based on the request/response approach. To communicate PLM data, it uses instances of the class PLMCoreContainer. To request data from the system, it uses instances of the class PLMQuery. The Message Connection service handles communication based on a message exchange approach. It queries and deletes messages from and writes messages to a service.

The next layer is something called the legacy plug-in manager, whose main purpose is to keep track of the different plug-ins which encapsulate legacy data. Another purpose of the plug-in manager is to keep track of which plug-ins handle which requests – for example requests about mechanical parts go to the CAD vault. This enhances the performance, since the alternative would be to ask each plug-in if it is responsible for each request, which would be hard to make scalable. The plug-ins themselves have a standardised interface with a set of methods which abstract certain functionalities. One such method abstracts the write functionality which is different for each legacy system. The method is called 'write' which has three parameters: filename, content and description. The filename states where to store an entry; it can be a URL for folder to store a CAD file on a server. Content is the actual data to be stored, the CAD file. Description is text describing the stored entry, for example metadata about the CAD file. Besides these methods, the plug-ins are also responsible for mapping information from the legacy system's specific information model to STEP AP214 used inside the service domain.

3.2 Connecting to the service-oriented architecture

When connecting applications in the top of Figure 3, the IT-related issues to resolve are how to create a mapping between the application's information model and STEP AP214, in case there are strong reasons not to use STEP AP214 in the application. Besides the mapping, it is important also to make an analysis of the applications' inputs and outputs and to make sure that placeholders for these exist somewhere in the legacy environment.

In the other end, when encapsulating legacy systems and adding new systems, it is important that the application has an accessible application programming interface (API). The API does not have to be open, but the legacy plug-in should be able to be established either by internal programmers or by external consultants. Establishing the plug-in means creating the mapping between the legacy system's information model and STEP AP214, and it also means establishing a connection between the generic methods inside the plug-in with the methods that the system uses for its internal operations.

4 Engineering change management

ECM is an important part of product development. Some of the early contributions focused mostly on empirical studies and description of how ECM has been arranged, along with its causes, characteristics of the changes, duration, resources, organisation, supporting tools and so on (Pikosz and Malmqvist, 1998). Huang and Mak (1999) conclude that ECM is a time- and resource-demanding activity and that, depending on the changes' extent, it can involve large parts of the company. In one study (Huang et al., 2003), over 80% of the respondents replied that cause of changes can be traced to poor communication and late discovery of problems. Eckert et al. (2004) have found two major sources for changes; these are the Emergent changes, which are caused as a result of refining the design, and Initiated changes, which are the result of external interference such as changed requirements. Lee et al. (2006) present a list of causes for ECs that, besides poor communication, include careless mistakes, snowballing change (changes due to other changes), cost savings, ease of manufacturing and product performance improvements. Lee et al. argue, however, that ECs are not always unnecessary. Many of them actually are beneficial and ECs should be viewed as inevitable, implying that efforts should be directed towards managing them rather than avoiding them.

Supporting tools for ECM have historically been mainly paper-based with occasional computer support, in which case the support is used for keeping track of the process and versioning of documents. Commercial PLM systems aim to use workflow and data management as support for ECM. Similar to this, Huang and Mak presented a set of requirements which mostly focus on computer support to provide control of the ECM process and ensure data availability and validity.

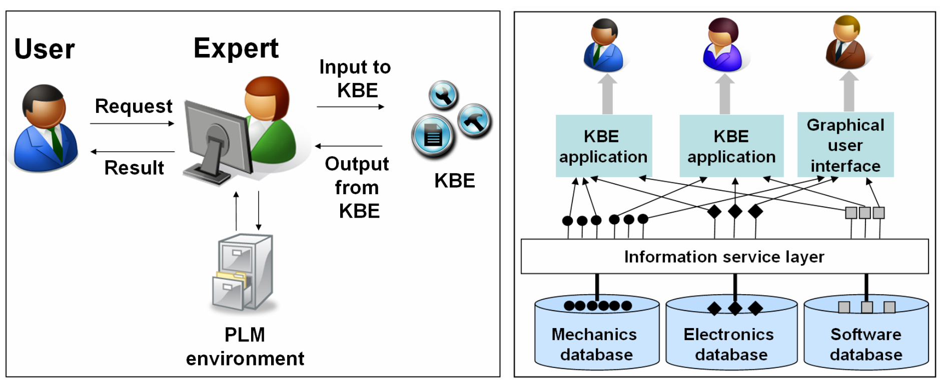
In more recent work, however, Joshi et al. (2005) conclude that only managing the workflow and versioning of part number identities, as ECM support in modern PLM systems does, is not enough since it does not take into account the fact that ECs are different from case to case. They propose a dynamic workflow driver implemented in the PLM systems that also captures and reuses knowledge about different ECs to continually improve the dynamic workflow driver. The idea of capturing and re-using knowledge generated during ECM was implemented by Lee et al. (2006). The knowledge refers to knowledge about the product which is captured and reused in order to help the decision-makers to predict change impacts.

Work that relates ECM with the underlying PLM architecture was presented by Bergsjö et al. (2007). It is concluded that whereas a one-system approach theoretically provides the best support for ECM since all the needed information is in one place and managed by one information model, this approach is impossible to realise in practice, partly due to different needs from different engineering domains and partly due to integration of different suppliers in the extended enterprise. The authors conclude that a more realistic approach is the application of a modular architecture with an integrating layer through which each local set of databases and applications is accessed.

4.1 KBE, ECM and Service-oriented PLM architecture

The general purpose of a service-oriented PLM architecture is to enable information integration in a loose and manageable fashion. SOA support for integration of KBE and PLM is illustrated in the right part of Figure 4, where the underlying information and database layer offer bits of information, provided as information services, to the applications which support users.

Figure 4 Left: current practice of integration of KBE via human interface; Right: service-oriented PLM architecture (see online version for colours)



4.2 Engineering change management supported by KBE

As mentioned in the previous section, ECM is a time-consuming and knowledge-intensive process. It is filled with different decisions which are in constant need of information and knowledge about the product affected by the change. Therefore, every decision is based on the decision-maker's knowledge. Similar to Lee et al. (2006) and Joshi et al. (2005), the authors of this paper have developed a method and a demonstrator that implements this method in order to illustrate the basic principles. Unlike Lee et al., the main focus of this contribution is to make the ECM process more efficient by application of existing knowledge and not so much on how to capture new knowledge. Joshi et al. propose that an EC can be evaluated by either people or applications whose main purpose is evaluation of ECs. In this work, we focus on a method which makes use of KBE applications whose purpose is to evaluate ECs, but also of those which originally had another purpose, such as design automation or analysis. The general idea is to reduce the time needed for a change evaluation and provide the change initiator with information about change impact. Effects of change impact on different configurations can be evaluated quickly, and only those affected can be involved in the resolution of the change.

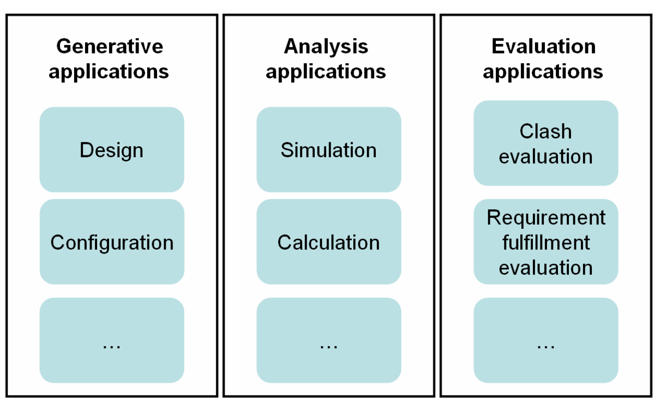
There are three different classes of applications which can support the EC process, illustrated in Figure 5. The first class is the generative applications. The data generated are then simulated in analysis applications, internally developed or commercial, which will analyse some product characteristics with the proposed change. The generative applications will typically have some other original purpose, but can be used to support the simulating applications. A typical example would be an application which configures the product so that only the configurations affected by a change are simulated. The main purpose of the simulating applications is to answer the question ‘Which are the new characteristics of the product given the proposed change?’ The simulated characteristics may be related to function, manufacturability, cost, and so on. The requirements for these simulating applications are:

1 *Parameterised*: in this context, parameterisation means that the simulation pplication uses editable variables in order to allow for parameters (or properties) to be changed.

2 *Integrated*: integration in this context refers to the way the application interacts with the PLM environment. The absence of hard-coded product parameters mentioned above means that the application needs to be provided with product parameters dynamically and in real time from the PLM environment.

After the analysis application is finished with defining the properties and the behaviour of the new product, the results need to be evaluated. This phase is more ‘engineering knowledge’-intensive, since conclusions are drawn about the new characteristics of the product. In a simple case, for example of a requirement fulfilment evaluation, of this application would only perform a comparison between the new product characteristics with the original technical requirements, such as maximum temperatures, pressures, velocities, cost, manufacturing time and so on, to see if the simulation result falls within accepted boundaries.

Figure 5 Three kinds of KBE applications for support of ECM with examples (see online version for colours)



4.3 Integration of KBE and PLM

In the work preceding the demonstrator, an investigation of how KBE applications are used to support the engineering process in current practice, it was concluded that KBE is weakly or not at all integrated in the PLM environment (Ćatić and Malmqvist, 2007). The most common situation is that the expert who developed the KBE application acts as the interface towards it; see left part of Figure 2. This means that the expert is responsible for providing the application with inputs, executing the application, and finally interpreting and storing the results. The human interface contributes to inefficiencies in the overall process, and if the user's request rendered an execution of the KBE application, overall process efficiency would increase. This would require the KBE application to have access to data and information stored in the PLM environment, and also that manual activities involved, for example interpretation of results, would need to be automated as part of the KBE application. There are several available approaches for integrating applications towards the PLM environment. In the context of this paper, a SOA was chosen due to its flexibility and modular view of the IT-architecture, supporting requirements on the KBE applications stated in Section 3, and since this architecture shows promising signs of supporting various industrial settings (Bergsjö et al., 2007).

5 Demonstrator development and evaluation

This section describes the two implemented demonstrators in more detail. Before this, however, there is a need to describe the product and the process background along with the legacy application environment.

5.1 Technical standards and techniques used for demonstrators

Regarding the technical aspects of the demonstrators, the SOA integration is based on the standard PLM services 2.0, described in Section 3.1. However, in order to create a fully functional demonstrator, other enabling technologies were utilised. The web application was implemented using Google Web Toolkit (GWT) which is a framework for building advanced AJAX web applications using the Java programming language. This gave the advantage of developing both the server and the client in a single development environment and yet taking advantage of techniques that range over a number of languages, from JavaScript and HTML on the client side to Java on the server side.

The legacy databases are modelled in excel-sheets, which are handled by excel legacy plug-in. The plug-in utilises the ODBC built-in support of Microsoft Excel to form a communication link between Java and Excel.

Building on top of HTTP, Simple Object Access Protocol (SOAP) provides the ability to access data and services over the web. It is also used to wrap information sent between provider and requester. This is accomplished by passing XML messages between requester and provider.

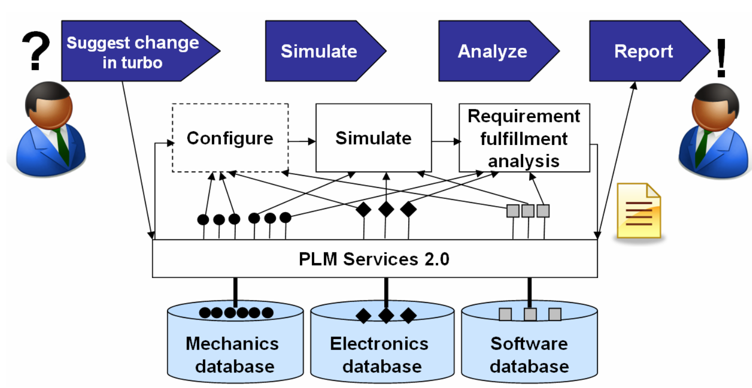
5.2 Turbocharger demonstrator

The case used for the demonstrator is a turbocharged diesel engine for heavy commercial vehicles. Developments of every new generation (or facelift) of the engine are performed in development projects. The ECM is governed within the project and most ECs are done during the development phase. The development itself begins with hardware development. When the hardware is frozen, electronics and software are developed in detail. Therefore, hardware changes are more or less forbidden after the freeze. This is, however, not always the case since critical changes can be made to the hardware even after freezing, for example if a large fault is detected, or if major improvements are identified. The issue of EC is administrated through the project management team in which each subsystem's responsibility is involved. Decisions are made by this team according to consensus on whether a change is to be approved or not. Decisions are made mainly based on experience. If a change is considered to need a deeper analysis, the project management team evaluates this by consulting experts. The turbocharger is a subsystem that has a high tendency and frequency of change proposals. This subsystem has a large effect on the overall engine performance and specifically on engine characteristics which are important to the end customer – torque, power, fuel consumption and emissions. Besides, the same turbocharger is used in several different engines, adding a variant management dimension to the EC process.

There is an analysis application that simulates the whole engine, developed in-house by an expert. This expert is consulted every time an engine simulation is requested. The expert executes the application, analyses the results and presents these to the simulation requesters. His application is hard-coded with product parameters for two major variants of the engine, and only he knows how the application is executed, how it is analysed and which data it needs from the PLM environment.

In order to support the above-mentioned engine development process, it was proposed for the demonstrator to evaluate how a change in one part of the engine affects other parts. The main purpose was to provide the decision-makers with better change impact analyses, and the designers with a possibility to test several alternatives to a particular change. The initial idea was to use the analysis application along with a configuration application which will provide the simulation analysis with correct inputs for the unchanged subsystems. Finally, the analysis would be performed by simply comparing the new characteristics with corresponding requirements. The communication of data and execution of applications would be performed through a service-oriented PLM architecture to ensure flexibility along with real-time access to the right data, accounting for possible changes that might have occurred in other subsystems. This process is illustrated in Figure 6.

Figure 6 Demonstrator ECM process and IT support (see online version for colours)

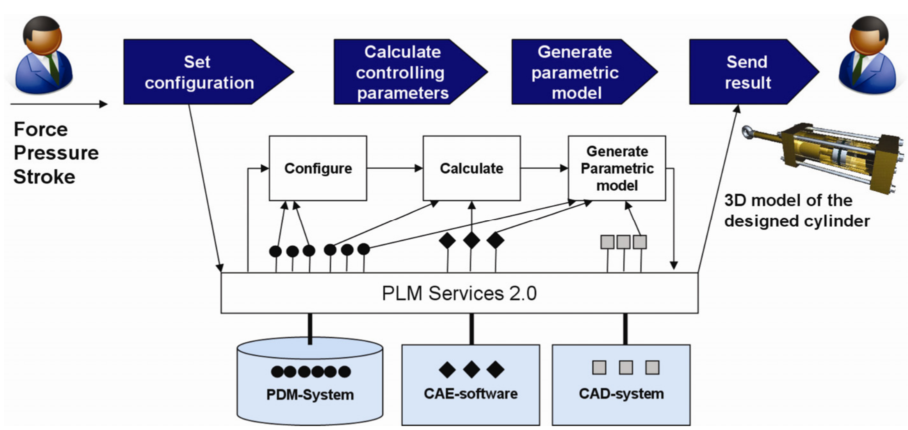


The approach proved hard to realise in its full scope due to the simulation application being hard-coded with product parameters for only one engine configuration. This made the need for configuring application obsolete (indicated by the dashed line in Figure 6). It also meant that changes in the complete engine could not be supported. The application was then used to provide parameterisation for the turbocharger subsystem. This meant that the simulation could only answer how changes in the turbocharger affect the rest of the engine. This reduction of scope was, however, acceptable from a business process point of view. The utilisation of a service-oriented PLM architecture was used to communicate data from different sources. When the analysis is done, unmet requirements and exceeded specifications are reported back to the initiator of the change.

5.3 Hydraulic cylinder demonstrator

The second demonstrator aimed to show a more integrated way to work with analysis and evaluation software, based on a parametric CAD model of a hydraulic cylinder system (Figure 7). The connection to CAD and visual information about the EC is one step to further automate the EC process. Also, software parameters to control the electronic system of the hydraulic cylinder are included in the EC, in order to demonstrate the integration of several disciplines in product development. Compared to the turbo demonstrator this is not based on a real industrial case, but it is intended to show possibilities of parametric design and analysis software that fully comply with the principles of integrated and parameterised product descriptions as described in Section 5. The EC workflow is initiated from the PDM system interface. In this interface, it is possible to view customer orders of different hydraulic cylinders. This interface collects data from a simulated legacy environment, consisting of different excel spreadsheets. As parameters are manipulated in the interface and saved as a result of a new customer requirement, an EC order is created and an e-mail is sent to inform all related persons. The e-mail addresses are accessed from a human resources (HR) database using a SOA interface. The change order triggers the start of a KBE application designed to identify changed customer orders. As the KBE application identifies a new configuration where an order has not previously been calculated, it collects all information needed for a calculation and sends the information to the CAE software located at a second site.

Figure 7 Data are collected and processed for a simulation of the changed product at Site 1 (see online version for colours)



First, a Matlab/Simulink script is triggered that calculates the new parameters, and the result is new geometrical dimensions for the hydraulic cylinder. In the next step, CAD software is automatically started and the parametric CAD model is changed according to the new parameters. Finally, the results of the automated process are packaged in a ZIP file that includes both CAD data and simulation results.

When the Change management KBE receives the data, the change order is completed and an e-mail containing information about the changes performed is sent out to the interested parties. The PDM system is updated and the change order is closed. If the simulation of the new parameters was unsuccessful, the parties are informed about the problems that require manual attention.

5.4 Discussion regarding process aspects

The applicability of the PLM Services 2.0 has been tested practically with the use case from the change management within a turbocharged diesel engine, and further elaborated with a fictitious hydraulic cylinder example. These demonstrators have been developed using OMG PLM Services 2.0, which has been shown to be a feasible standardisation effort, especially when considering the alternatives such as supplier single storages and software suites.

It is relatively easy to adapt PLM services to your internal systems, but it is more difficult to ensure integration within the extended enterprise. The PLM services interface between customer and supplier is not fully developed. PLM services is more focused on internal exchange of product information, for example from product development to production.

The usability benefit of a SOA is basically that engineers would continue to work with the applications they like, but at the same time get customised services for performing time-consuming information management tasks. Traditionally, processes are based on people and on sending notifications to the right person. However, the multidisciplinary integration of ECM is only one part of the problem. This study showed that the development could be more efficient due to the fact that many process steps involving manual input of information can be automated with the help of KBE applications. Future studies will have to show the applicability of such integrated tools in an industrial setting. There is often skepticism about process automation, but the work presented here uses the KBE application to act upon the request of a human, and is restricted by boundaries to align within predetermined boundary conditions. It is thus not the KBE application that makes decisions and verifies its own work.

Our main focus in this work has been on existing KBE applications which are already implemented but not integrated with the business processes. The reasons for this delimitation are three. First of all, the fact that they are implemented in some sense means that someone in the business process is aware of their existence. The second reason is that our focus is on supporting the ECM process and not on developing KBE applications. The third reason is that once we implement some KBE support for the ECM process, it is easier to justify new investments in this kind of support. The showcase provides a good basis for the strategic discussion about which parts to support next, and through which information and applications. The alternative is a 'big bang' approach with many big solutions, which leads to issues concerning organisational change management, knowledge capture and development of new applications. All of this increases the risk that people feel threatened by increased automation of tasks, along with the usual friction of introducing new ways of working along with new applications (KBE or not). The authors perceived an incremental process of introduction and small successes over a period of time as a more promising introduction strategy.

In the case of ECM, it is important that the integration between KBE and PLM allows appropriate access and even concurrent access to information as the information is changed, so that lags in product development do not occur. This is evident when development is performed in close collaborations, such as in-house development where different departments need to collaborate and balance different characteristics of the product. For these types of collaboration, and in contexts where the end users do not use the same PLM system, the SOA is likely to be the most efficient means to integrate KBE. Rigid integrations are in most cases quick to perform but are difficult to maintain over time when software is upgraded and changed. This is when the existence of a contract such as the one delivered by OMG for PLM Services 2.0 greatly facilitates the integration.

5.5 Discussion regarding IT and architectural aspects

When reflecting upon the study and the research questions, it can be said that one of the practical difficulties with the PLM services standard is that it lacks detail. For example, a couple of queries were added in order to manage EC requests and orders. This implies that the standard had to be expanded to support the case we tried out. It means that the current service layer is not fully covered by the standard, and integration to other PLM services 2.0 service layers is not likely to work right out of the box.

An extended implementation guide would be beneficial in order to more quickly start to work with the implementation of the standard. OMG has chosen not to specify the interfaces, which implies difficulties in order to use PLM services in the extended enterprise.

Documentation is sparse throughout the standard. In the beginning, it was difficult to ensure required information (required annotation). Good knowledge in AP214 and expert programming skills (preferably in Java) are a prerequisite for understanding the implementation fully. The unspecific standard would make it possible for different dialects of the implementations to emerge, which is not preferable when different SOAs are being integrated.

One of the main advantages with SOA as discussed in this paper is that it enables every company to customise and standardise the IT environment using a loose integration concept that would simulate a single storage towards the user. The users are not forced to work directly towards the database layer, but are working through their ordinary GUIs and applications towards the service layer. Problems with loose integration that have to be managed are aspects related to more complicated governance and maintenance functions. The services exist in a layer outside the traditional PDM/PLM system, which makes it more complex to manage. Related research (e.g. Bergsjö et al., 2007; Lee et al., 2007; Zimmerman, 2008) as well as this research has shown that administrative tools with management and documentation capabilities need to be developed to support both the implementation and governance processes of service-oriented PLM systems.

In traditional supplier-focused SOA, suites and single-source solutions have often been attempted to lock in the customer to use applications and systems from one single supplier. A different approach has been an attempt to duplicate information in a new location that later can be accessed in a standardised fashion. These so-called hub solutions show, instead, data redundancy and data integrity problems. These different types of supplier lock-ins have in reality meant that companies would have to change their way of doing business in order to integrate and share information within and across the extended enterprise. With an open SOA standard, this can be avoided. And with a large effort like OMG PLM Services, it is going to be possible to influence the IT suppliers to comply with the new standard, or even to keep legacy systems or develop new internal PDM/PLM systems that can communicate with external systems through the services they supply.

A problem with this standardisation effort, as with many similar efforts, is its future use as an industry standard. We believe that OMG's effort here is a possible candidate especially in the automotive industry. The fact that it is based on ISO/STEP AP214 and being developed by and for especially the German automotive industry is a good sign that it will be used in the future. SOA is one of many different architectures for achieving efficient multi-domain integration. There are, however, alternatives that have been used effectively in the industry, for example Bergsjö et al. (2006). For example, in hub solutions like Zimmerman and Malmqvist (2007) where information can be published and retrieved over the entire product life cycle, there is also the possibility to use advanced access right scripts and tools that can be used to give suppliers and customers access to the company's PLM system. Competing standards, when regarded from a SOA perspective, are those that are being developed by large IT-suppliers; IBM and Oracle are suppliers that genuinely seem to be developing their own open standards for SOA focused on their particular fields, for example finance human resources and enterprise resource planning systems. The disadvantage of those giants is that they are not the big players in the PLM field, and the future will have to show how suitable their solutions are to work with, for example, engineering tasks and, more importantly, engineering tools.

6 Conclusions

In this paper, it has been shown that a SOA can benefit both user and business perspectives of PLM. These ways include, but are not limited to, issues regarding PLM architecture, control of the business logic and superior usability. The applicability has also been tested practically with the use case from change management in a turbocharged diesel engine, and a fictitious example involving the EC and change impact analysis of a hydraulic cylinder. This demonstrator has been developed using OMG PLM services 2.0, which has been shown to be a suitable standardisation effort. The framework has been shown to be applicable to support ECM along with two developed KBE applications that simulate effects of a change in real time, as the product is updated in the PLM system.

PLM architecture is improved since a SOA allows transparency and flexibility in IT integration, whereas supplier suites and single-source solutions actively work against this principle. In a SOA that is based on an open standard like OMG PLM Services 2.0, the principles of a SOA of modularity, central coordination, standard communication, general modelling constructs and minimum process redundancy can be managed.

The control of the company's business processes means that the company does not outsource the way it is doing business to an IT supplier, who does not necessarily understand the requirements in a particular business. The service-oriented PLM architecture allows for flexible integration of the current business processes, and instead puts demands on IT suppliers to support standardised interfaces rather than forcing every company to work according to their PDM system logic.

Superior usability is achieved since information services are created focusing on a specific need of an engineer or a development process. These services do not change the way people are used to working with the applications, but rather add a new service layer for those who benefit from it, and those are most likely engineers and managers working cross-functionally with new and innovative products. It is concluded that a service-oriented PLM architecture is an efficient IT architecture that enables multidisciplinary integration and collaboration. In this context, it is also concluded to be the most promising architecture to support ECM.

OMG PLM services need to be improved regarding documentation, a higher degree of support for the implementation, better organisation and a higher detail level. This is particularly important for communication within the extended enterprise, where a transparent implementation of the standard is a prerequisite in order to enable different implementations of the standard to communicate with each other.

Our future work includes work with a focus on general IT architecture – focusing on governance functions as well as modelling and maintenance issues with a PLM architecture and the integration of both service-oriented and legacy PLM architectures. It would also be interesting to further study the application of a SOA in an extended enterprise context.

References

Abramovici, M. and Bellalouna, F. (2008) 'Service oriented architecture for the integration of domain-specific PLM systems within the mechatronic product development', in *Proceedings of TMCE 2008 Symposium*, Izmir, Turkey.

Bergsjö, D., Malmqvist, J. and Ström, M. (2006) 'Architectures for mechatronic product data integration in PLM systems', in *Design 2006*, Dubrovnik, Croatia.

Bergsjö, D., Vielhaber, M., Malvius, D., Burr, H. and Malmqvist, J. (2007) 'Product lifecycle management for cross-X engineering design', in *ICED'07*, Paris, France.

Burr, H., Vielhaber, M., Deubel, T., Weber, C. and Haasis, S. (2005) 'CAx/engineering data management integration: enabler for methodical benefits in the design process', *Journal of Engineering Design*, Vol. 16, pp.385–398.

Ćatić, A. and Andersson, P. (2008) 'Manufacturing experience in a design context enabled by a service oriented PLM architecture', in *Proceedings of the ASME DETC*, New York, USA.

Ćatić, A. and Malmqvist, J. (2007) 'Towards integration of KBE and PLM', in *ICED* Paris, France.

CIMdata (2006) 'PLM and ERP integration: business efficiency and value', Ann Arbor, Mi, USA.

Dahlén, M. (2006) 'From PDM to PLM at Autoliv', in *Nord PLM 06* Göteborg, Sweden.

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

Feltes, M. and Lämmer, L. (2007) *Introduction to PLM services 2.0*. Available at: http://www .prostep.org/en/standards/plmservices/ (accessed on 24/11/2007).

Georgiev, I., Ovtcharova, J. and Georgiev, I. (2007) 'Modelling web services for PLM distributed system', *Int. J. Product Lifecycle Management*, Vol. 2, pp.30–49.

Hallin, K., Zimmerman, T. and Malmqvist, J. (2004) 'Towards a framework for integrated information management in mechatronic product development', in *Proceedings of the NordDesign 2004*, Tampere, Finland, pp.153–162.

Huang, G.Q. and Mak, K.L. (1999) 'Current practices of engineering change management in UK manufacturing industries', *Int. J. Operations and Production Management*, Vol. 19, pp.21–37.

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

Joshi, N., Ameri, F. and Dutta, D. (2005) 'Systematic decision support for engineering change management in PLM', in *Proceedings of DETC/CIE*, Long Beach, CA, USA.

Lee, H.J., Ahn, H.J., Kim, J.W. and Park, S.J. (2006) 'Capturing and reusing knowledge in engineering change management: a case of automobile development', *Journal of Information Systems Frontier*, Vol. 8, pp.375–394.

Lee, T., Lim, J., Shin, J., Myung, S., Choi, M., Baek, S., Kim, J., Oh, J., Lee, D. and Han, Y. (2007) 'An implementation methodology of SOA based PLM system', in *PLM'07*, Bergamo, Italy, pp.303–310.

MacKenzie, C.M., Laskey, K., McCabe, F., Brown, P.F., Metz, R., and Hamilton, B.A. (2006) *Reference Model for Service Oriented Architecture 1.0*. Available at: "http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm (accessed on 2009-06-06). Committee Specification 1, 2 August 2006.

OMG (2006) *Object Management Group*. Available at: http://www.omg.org/ (accessed on 24/11/2007).

Pikosz, P. and Malmqvist, J.A. (1998) 'Comparative study of engineering change management in three Swedish engineering companies', in *Proceedings of DETC'98*, Atlanta, GA, USA.

Srinivasan, V., Lämmer, L. and Vettermann, S. (2008) 'On architecting and implementing a product information sharing service', *Journal of Computing and Information Science in Engineering*, Vol. 8, pp.1–11.

Svensson, D. (2003) *Towards Product Structure Management in Heterogeneous Environments*. Göteborg: Chalmers University of Technology.

Vec-Hub (2007) *Virtual Enterprise Collaboration Hub (VEC-Hub)*. Available at: www.vec-hub.org (accessed on 24/11/2007).

Zimmerman, T. (2008) 'Implementing PLM across organisations – for multi-disciplinary and cross-functional product development', in *Product and Production Development*. Vol. Doctoral Thesis Göteborg: Chalmers University of Technology.

Zimmerman, T. and Malmqvist, J. (2007) 'Facilitating interdisciplinary PLM through usage of hub methodology and 3D lightweight visualisation data', in *Proceedings of ICED'07*, Paris, France.

Paper C

Manufacturing Experience in a Design Context Enabled by a Service Oriented PLM Architecture

Ćatić, A. and Andersson P. (2008) Proceedings of DETC08: ASME Design Engineering Technical Conferences – Design for Manufacturing and the Lifecycle Conference, Paper ID DETC2008-49858, 3-6 August, New York City , NY, USA

Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Design for Manufacturing and the Lifecycle Conference
IDETC/DFMLC 2008
August 3-6, 2008, New York City, NY, USA

DETC2008/49858

MANUFACTURING EXPERIENCE IN A DESIGN CONTEXT ENABLED BY A SERVICE ORIENTED PLM ARCHITECTURE

Amer Ćatić
Product and Production Development
Chalmers University of Technology
amer.catic@chalmers.se

Petter Andersson
Functional Product Development
Luleå University of Technology
petter.andersson@ltu.se

ABSTRACT

An increased competition on the product development market pushes the industry to continually improve product quality and reduce product cost. There is also a trend towards considering a products life cycle aspects including environmental sustainability. The manufacturing process is a major cost driver in the product life cycle; hence, there are many initiatives to improve manufacturability and reduce production cost. Learning from earlier projects is essential to avoid recurrence of problems and is generally realized through use of concurrent engineering and design for manufacturing (DFM). Other research provides general DFM principles which state detailed guidelines for how different geometries combined with different manufacturing processes affect component quality and cost. The real competitive edge lies however in the development and application of company specific DFM principles that are based on manufacturing experiences. To do so requires an overview of and access to the collected manufacturing experiences. The aim of this paper is to point out key enablers for efficient reuse of manufacturing experience, which is considered to contribute to lower product cost and higher product quality.

A study performed at an automotive and at an aerospace engine manufacturer pointed out the apparent need and lack of reuse of manufacturing experiences in product development. Applications supporting reuse of manufacturing experience through embedded DFM knowledge in designer's CAD system were found in the literature. The issue of integrating these applications with the enterprise environment, in order to capitalize on existing sources of manufacturing experience, is addressed with a proposed solution applying a service oriented PLM architecture. In addition, a graphical user interface visualizing the manufacturing experience in a combined product and process context was developed. The validation of these proposed and developed solutions was done through interviews and workshops. The conclusions are that visualization of manufacturing experiences in a combined product and process context provides improved understanding of how the experiences relate to each process history and that a key enabler for integration of information in heterogeneous environments is the use of standard service oriented architectures and neutral formats.

1 INTRODUCTION

Globalization and intensified competition in the industrial world call for improvements in all Product Development Life Cycle phases and cross-disciplines. In the automotive and aerospace industry, the manufacturing process is in most parts a well integrated part of the Product Development (PD) process and it is still common with a collocated manufacturing shop floor. In this perspective, it is natural to include resources from the manufacturing functions in the earlier PD phases as a mean to share manufacturing experience from earlier projects. A pilot team in the concept phase is then assembled of expertise from other organizational functions such as design, CAE, market planning and sourcing. The shortcomings of such an approach becomes obvious in a large company with several stakeholders and where the knowledge base from manufacturing alone consists of up to 8 key persons. In addition to the difficulties to compose such a large group and the implications of keeping it efficient, the lack of resources is often a show stopper/obstruction. In order to provide access to e.g. knowledge and experience from manufacturing without having personal presence from the experts there is the possibility of developing expert and knowledge based computer applications which perform the most common expert tasks directly on the designer's desktop. A problem however is that these applications' input data is usually stored in a database specially developed to suit a specific need locally in manufacturing. Another problem is that there exists manufacturing experience that is explicitly available but is not suitable for a computer application e.g. comments regarding the design or regarding a workaround in the manufacturing process in order to realize a complicated design or other documents such as incident reports from manufacturing. This experience is very valuable for those working in earlier product phases in order to avoid mistakes and complications from earlier projects.

Hence, the aim of this paper is to address two key enablers for efficient reuse of manufacturing experience. These key enablers are:

- Access to manufacturing data
- Contextualization of manufacturing data from the receiver's point of view

In section 2 a previous case study to investigate current practices for reuse of manufacturing experience are described. In section 3 some of the most relevant work related to this contribution is summarized. In section 4 the concept of a service oriented PLM architecture is explained. Section 5 describes how the accessed experience is made available to the designer by putting it in a combined product and process context, thus making it more understandable. Section 6 describes why a service oriented PLM architecture enables the proposed contextualization better than currently applied solutions. In Section 7 the demonstrator that will implement the results from section 4 and 5 is described and explained in detail. Finally in section 8 a discussion around the subject and future work are proposed and the paper is concluded in section 9.

2 STATE OF THE PRACTICE

Prior to this work, an empirical study was carried out at one automotive and one aerospace industrial company [1], [2]. The aim of the study was to understand the current practices for capture and reuse of experience, i.e. engineering knowledge, in manufacturing. The study revealed a heterogeneous environment with several different sources of manufacturing data. Examples of sources for experience were;

- Lessons learned database for design engineers in Lotus Notes
- Best Practices database for design engineers on the intranet
- Global database for standardized manufacturing processes for manufacturing engineers
- Lessons learned database for manufacturing operations on the intranet
- Experience reports from concluded projects on local file areas or in physical folders
- Database with in-line measurements used primary by manufacturing operations
- Database for tracking problems in manufacturing.

The study showed that the automotive company has a more developed system to manage experience and are more

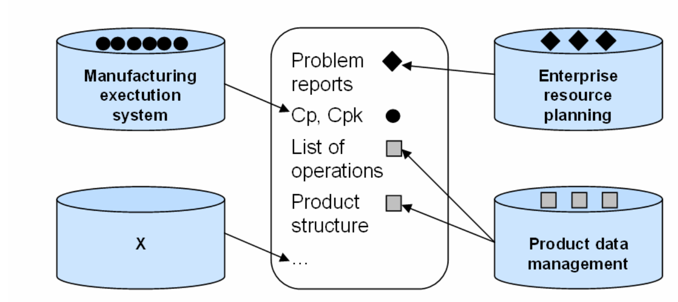


Figure 1 – heterogeneous environment for storage of manufacturing experience

aware about their processes for capturing manufacturing experience. Despite this, Manufacturing Operations convey a higher frustration over recurrent manufacturing problems. One possible explanation could be that an increased awareness of the complexity of problems increases the receptivity and also the motivation to solve the problem.

The study also revealed that although systems for capturing manufacturing experience existed within the manufacturing organization, the knowledge of its existence or how to access the information was not common knowledge among design engineers.

A set of requirements on a design system that integrates experience use was identified;

1. Need to interactively search, find, retrieve and integrate experience related information from several different sources
2. Need to keep the experiences up to date. As close to real time update as possible
3. Need to integrate in the designer's context and expand functionality of existing tools rather than building a completely new tool

3 RELATED WORK

The paper applies results from several different areas which relate to the general subject. Related work from each of these areas is briefly summarized below.

3.1 Knowledge based engineering and DFM

The integration of manufacturing experience and knowledge in product development generally referred to as design for manufacturing (DFM), is a well established approach for increasing the manufacturability and quality and at the same time decreasing the costs of the designed products. General DFM principles which state detailed guidelines for how different geometries combined with different manufacturing processes affect the component quality and cost can be found in e.g. [3].

Examples of implementation of DFM in internally developed Knowledge Based Engineering (KBE) techniques has been used as an approach to provide manufacturing knowledge in early development phases [4], [5], [6], [7]. The focus of these examples is to demonstrate different ways of incorporating manufacturing knowledge and experience through design automation and usually require manual handling of inputs and outputs.

There is however a need to provide these kinds of applications with inputs derived from e.g. databases containing results from manufacturing processes, which today are used for quality management purposes, in order to feed product design with accurate and up to date information from manufacturing.

An area approaching this is data mining where intelligent tools for extracting useful information and knowledge have been developed but the context of usage in a designer's context remain.

When simulating the manufacturing processes in order to obtain desired product properties in the final product, it is essential to include the entire manufacturing process sequence in these simulations [8]. In this paper we argue that it is equally important to include the full manufacturing process sequence when feeding back manufacturing experience in the early phases of product development.

Molina et. al. [9] demonstrates a system that utilizes web-based applications to, at the concept level, allowing a designer to describe a part so that an expert system can decide which manufacturing processes can produce the desired part, in the desired time, with the desired quality.

Other work on reuse of manufacturing experience is done by Alizon et. al. [10], presenting a method that considers similarity, efficiency and configuration when identifying similar existing designs to a desired one defined by the engineer.

3.2 Service oriented PLM architecture

Service oriented architecture as a software engineering principle has been around for many years but it is only recently with the increased maturity of web service technology that this kind of loose integration has been applicable. With rising insights regarding IT support of engineering processes especially related to issues of product documentation and the supplier lock-in phenomenon the principle of service orientation has been abstracted from basic software principles to integration of systems. The purpose of this is the fact that the product is documented in different systems containing bits of information about the product and in order to obtain a complete view of the product these information bits need to be gathered which means the underlying systems need to be integrated [11], [12]. From this perspective the systems are viewed as providers of information services which deliver these information bits. These ideas led to the development of a standard for how design of these abstracted services is to be implemented called PLM Services 2.0 and provided by OMG [13].

An implementation of the standard has been performed and the results seem very promising [14]. There are several works done which describe the possibilities of improving different parts of product lifecycle management through the application of a service oriented architecture [15], [16], [17], [18], [19], [20]

There are also other proposals for the realization of a service oriented PLM architecture, one of which is proposed by another standardization body called OASIS [21]. An implementation of this standard is found in the European VIVACE project [22] in a demonstrator for supporting the idea of an extended enterprise using a hub solution [23] that applies web services according to the OASIS standard for integration with other systems.

3.3 Contextualization

The importance of a contextual approach is widely recognized within Knowledge Management and the emerging field of IT/web collaboration tools.

The definition and use of context as a concept has been analysed by Bazire et. al. [24]. From there analysis of 150 definitions a few key parameters were identified like constraint, influence, behaviour, nature and system.

Context aware applications, as defined by Dey and Abowed [25] use context to provide task-relevant information and/or services to a user. The context is here primarily of four types; location, identity, time and activity.

In the European project VIVACE [22], a context based search platform was developed [26]. As part of the study, two approaches regarding a context model were studied, a top down approach and a bottom up approach. In the top down approach is the engineering context defined as any information that can be used to characterize the situation of an engineer. The bottom up approach deals with the problem of categorizing data/information and the recognition of new circumstances where the knowledge source could be usefully applied. A key issue concluded here is the importance of providing the right knowledge to the right user at the right time in the design process.

4 SERVICE ORIENTED PLM ARCHITECTURE

A service oriented PLM architecture implies that every source of data and information is viewed as an information service provider [11]. This is illustrated in Figure 2 where every information source publishes its available information services in a service registry. The registry is accessed by the user applications to search for the information they need. The service registry then appoints the user applications to the correct address of the information as published by the information sources. The information access and delivery is

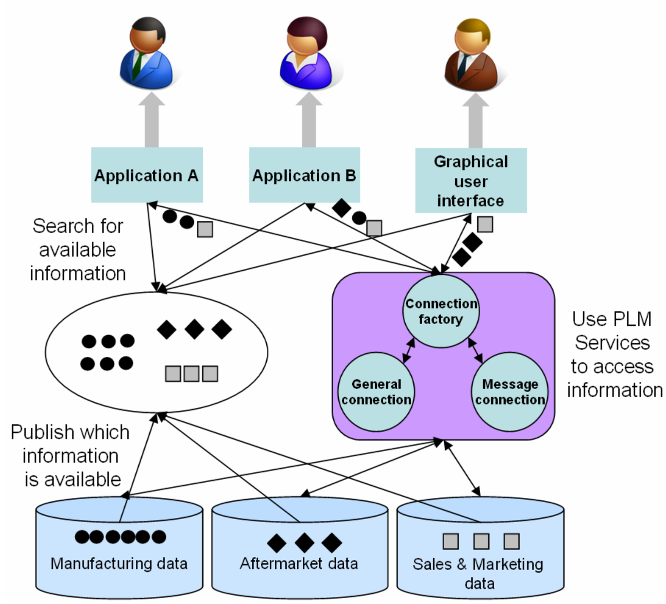


Figure 2 – Service oriented PLM architecture

then performed according to a contract that states how the information is accessed and delivered. The contract also states in which format information is delivered. In this case study a standard contract for the services, called PLM Services 2.0, is considered.

The primary purpose of a service oriented PLM architecture is to make sure that all the data gathered and stored throughout the product lifecycle is made available and can easily be accessed for different purposes. These purposes can vary; examples could be development of applications which apply aftermarket knowledge to analyze a packaging solution of a vehicle, calculate exact production cost for a given design, provide better support for strategic product portfolio decisions, provide analyses of material suppliers in real time for purchasing and so on. Even though all these examples could be realized by developing special databases for each purpose most of the time the data needed already exists in some form and in some database. If there is business value in developing an application such as the ones mentioned in the examples this value should not be decreased or wasted due to the fact that data needed for the application is difficult to access. This is depicted in Figure 2 where the underlying information and database layer offers bits of information, provided as information services, thus making information accessible for the applications which support users.

4.1 PLM Services 2.0

An important enabler for service oriented PLM architecture to work is the "service contract" according to which information is communicated. The standard contract PLM Services 2.0 is provided by the standardization body Object Management Group (OMG) [13] and has been developed together with representatives from the German automotive industry. PLM Services 2.0 standard provides the developer of the service oriented architecture with the contract according to which information is to be communicated.

What makes this standard special is that it's starting point are the common workflows encountered in the PLM area and its aim is to support engineers working with product development. This is however not to be confused with processes and workflows which are embedded in commercial PLM software suites due to the fact that the workflows in the standard are at a more generic level (due to the fact that they are not restricted to the use of any particular software for PDM nor for CAx). This means that there is flexibility to have company specific processes, applications and information.

The standard defines a STEP AP214 compliant data model and all the necessary functionality to realize several use cases. OMG supplies the XML schemes and WSDL (Web Service Description Language) files that define PLM Services. The WSDL files supplied by OMG specify three web services, Connection Factory, General Connection and Message Connection. The Connection Factory service contains method skeletons that handle authentication and the creation of sessions and acts as a gateway to the other two services. The General Connection service includes method skeletons that handle communication based on the request/response approach. To pass PLM data, it uses instances of the class PLMCoreContainer. To request data from the system, it uses instances of the class PLMQuery. The Message Connection service includes method skeletons that handle communication

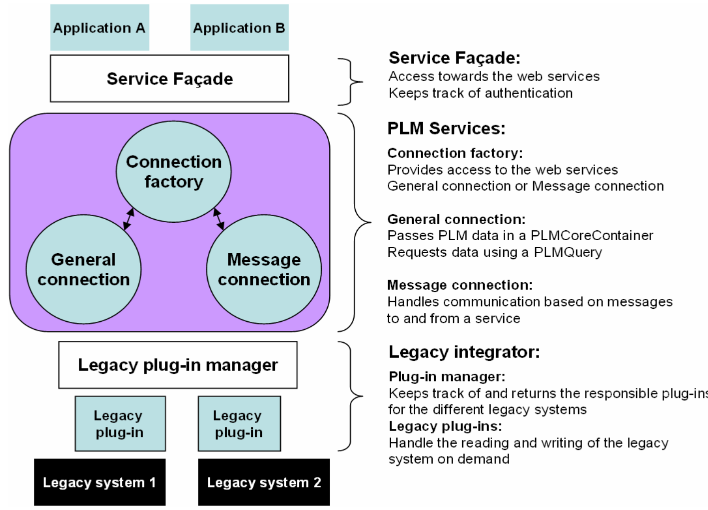


Figure 3 – PLM Services

based on the message exchange approach. It provides methods to query messages from a service, to write messages to a service and to delete messages from a service. The service layer setup is depicted in Figure 3.

5 CONTEXTUALIZATION OF INFORMATION

Providing access to data and information is an important and necessary first step but not always enough to support the processes in an effective manner. This only addresses the service oriented integration part of SOA in which the information sources are integrated with each other but there are no considerations of how the processes are integrated with the information. When the information sources are integrated possibilities and needs arise to change the processes in order to optimize the complete process and IT environment. Since a service oriented integration enables access to more information which has its origin in company departments who have another view of the product it will have another format due to the differing context. In the particular case of reuse of manufacturing experiences in design the information is created and stored in the context of manufacturing and thus it is formatted to support manufacturing needs. Therefore the information needs to be put in a context so that the receiver of the information is able to understand it in order to support the process the receiver works in.

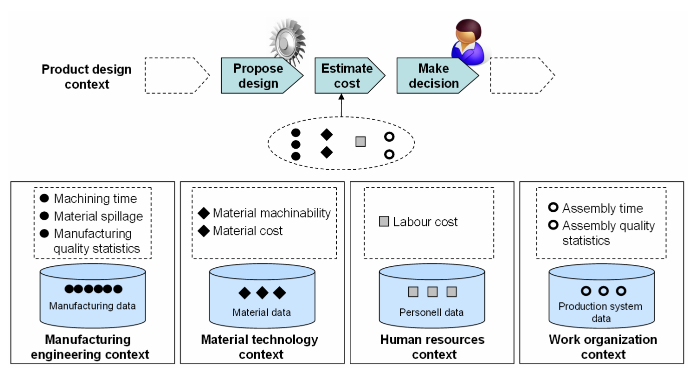


Figure 4 – Example of information from different contexts needed in a process which is in another context

Two issues regarding the contextualization of information have been identified:

1. Information format
2. Information presentation

The issue of information format refers to the fact that the information is formatted in order to support a process in its original context. This means that the information needs to be reformatted in order to support another process in another context. The issue of information presentation is simply the way in which the information, once accessed and reformatted, is presented to the user in order to support the user's process. The two issues related to contextualization can be addressed in several ways, two of which are:

1. Presenting the accessed information in a specific graphical user interface (GUI) which is suited to the user's context. The specific GUI implies that the information needs to be formatted in the way which is required by the GUI design or format. This solution implies that the information is presented in a logical manner to the user but the user needs to execute the process for which the information is needed.
2. Implementing a specific application which will use the accessed information as input, perform the process which the information supports and present the result to the user. This solution implies that the application needs to understand the format in which the information is accessed.

Which of these solutions is better to choose depends on the context and the process which is to be supported.

5.1 Product and Process context

In this case study the issue of contextualization has been addressed by applying a combined product and process context. To only use the product structure, which ever structure it may be, to structure experience is suitable for e.g. design guidelines but lots of experiences relate to specific activities during design, manufacturing, sales, service etc and therefore the process aspect is needed as well. When considering experience related to a component it will be part of a system that performs a function, taken from the designer's view of the product, but the component will be part of a subassembly, taken from the manufacturing engineer's view of the product. This issue of different product views is addressed by e.g. chromosome model [27] which can be used to bridge the two contexts by relating two different product structures. The issue of process related experiences remains however unsolved.

In the particular case of reuse of manufacturing experience the experience is related both to the manufactured components which can be structured in assemblies viewed from the manufacturing point of view. But it also relates to the different steps of the manufacturing process. Therefore there is a possibility to select a component and reach all the experience related to this component from all the manufacturing steps. By applying the process context onto the product context it is possible to also select an activity in the manufacturing process and only view the subset of experience related to the selected component and selected manufacturing step.

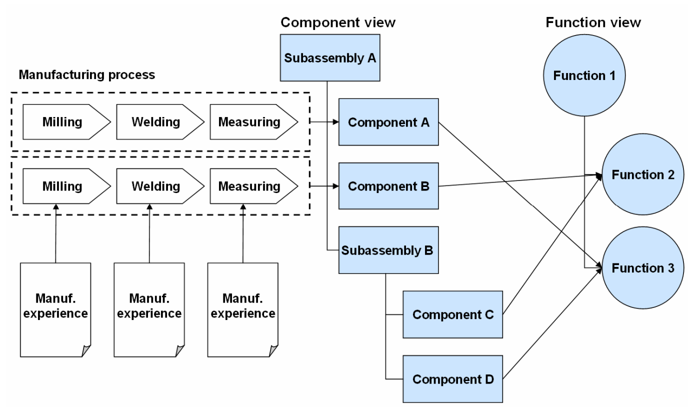


Figure 5 – Bridging experience from manufacturing to design context

Finally there is also a possibility to only consider a manufacturing activity, e.g. welding, and not consider any components in which case the experience will relate to welding in general.

Completing this with the chromosome model which relates components with functions we come even closer to the designer's context. This means there is a possibility to e.g. view all experience of welding related to a specific function which enables the bridging of manufacturing experience from the manufacturing to the designer's context as depicted in Figure 5.

6 SERVICE ORIENTED PLM ARCHITECTURE AS ENABLER FOR CONTEXTUALIZATION

In this section the issue of using a service oriented PLM architecture for enabling contextualization is compared to the state of the practice enablers. The example from Figure 4 will be used to illustrate and discuss the differences of the described solutions. In the example a fictive process of cost estimation is supported. From a process point of view it will be best supported by implementing a cost estimation application which needs the listed pieces of information as input in order to produce an estimate as output. Thus the contextualization of information from four different contexts to the design context is performed by a specific application. The application will present the final information, the cost estimate, in a way and in a format which is best suited from the designer's point of view.

In Figure 6 a common state of the practice is described. A cost estimation expert either designs an application himself or helps an application designer to design a cost estimation application. The application is designed by hard coding the different pieces of information into the application. This usually leads to issues regarding the fact that it is costly and time consuming to develop different applications to support the development of different product variants why either the most common variants are supported or the hard coded parameters are balanced and their values are approximate. The hard coded parameters need to be updated after a while and the application needs to be maintained.

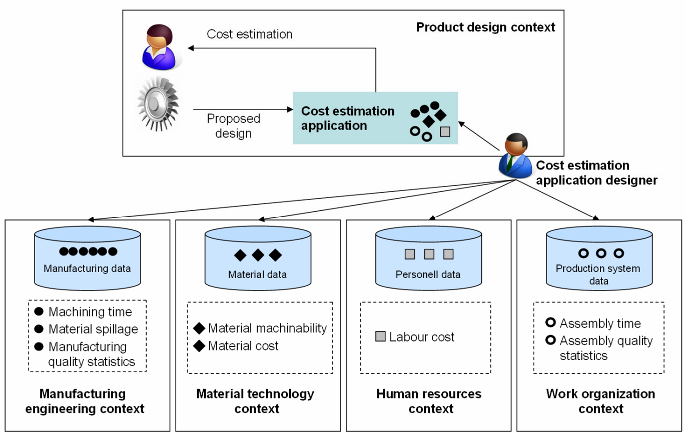


Figure 6 – Contextualization through hard coded application

These circumstances lead to increased costs for the development and maintenance of the application which means that the overall financial gain in the process is decreased. The application costs might even be so high that the financial gain is lost and the application is not implemented even if it may increase product quality which is hard to measure exact financial gains from.

Figure 7 depicts another solution which is about creating interfaces in order to integrate information sources and applications. This approach addresses the issue of not having to balance different parameters due to product variants in the application which means the process will be supported in a better manner. The approach will however lead to a lot of hard coded integrations with many to many integrations which themselves increase maintenance costs when IT vendors change their interfaces in new releases. The flexibility of changing processes is decreased and the changes with minor financial gains will not be implemented due to interface development costs.

In Figure 8 a service oriented PLM architecture is used to enable the contextualization. The cost estimation application accesses the information service layer and requests the information it needs for its process of cost estimation. This approach addresses the issue of not having to balance the parameters as is the case in the hard coded application in Figure 6. At the same time the coupling to the underlying information sources is not either hard coded through direct interfaces as in Figure 7. The loosely coupled SOA approach

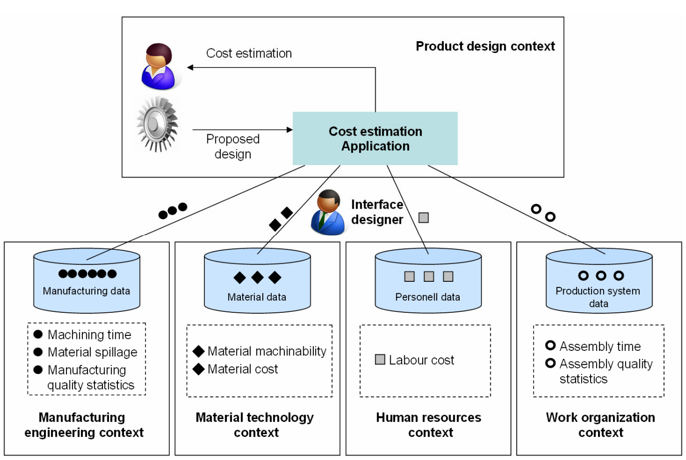


Figure 7 – Contextualization through hard coded interfaces

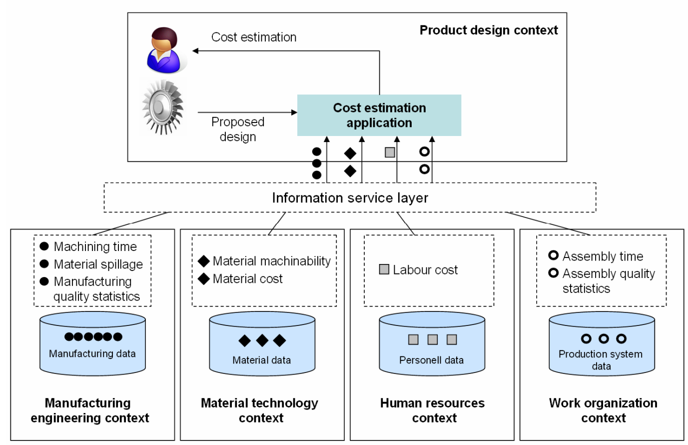


Figure 8 – Contextualization through a loose SOA integration

does however imply that the information needs to be in a neutral format according to the service contract. This is required in order to provide the needed flexibility since a neutral format will mean that every information source and every information consumer will only need to have one interface which is needed to deliver/access the information. The most optimal approach is to use an information standard which is supported out of the box by the information sources. This is also the case in the PLM Services 2.0 standard which supports the information standard AP214 that is also supported by most commercial PDM systems.

7 DEMONSTRATOR

The aim of the demonstrator is reuse of manufacturing experience in early design phases. The general purpose for why this focus is chosen has been addressed in sections 3 and 5. The studied case contains all of the issues that have been described so far. More explicitly these are:

- The manufacturing experience considered is stored in four different systems.
- The format of the information carrying the manufacturing experience is adapted to the manufacturing context, not design.
- Once accessed and reformatted the information needs to be presented in a way which is natural and logical from the designer's point of view.

Schematically the demonstrator architecture is depicted in Figure 9. In Section 5.1 it was described how a product and process context was used to create a bridge for manufacturing experience from the manufacturing context to the designer's context. Technically this will need to be done by implementing a neutral information model in the service layer. This provides the desired flexibility that is one of the main reasons for choosing a SOA. This means that all the information sources and information consumers need to be able to communicate to the neutral format.

The manufacturing experience consists of measurement data stored in a legacy system, production preparation documentation stored in Siemens TeamCenter, operator comments stored in a legacy system and incident reports stored in SAP R/3. To cope with these issues there is a special process that states the order and type of the different queries

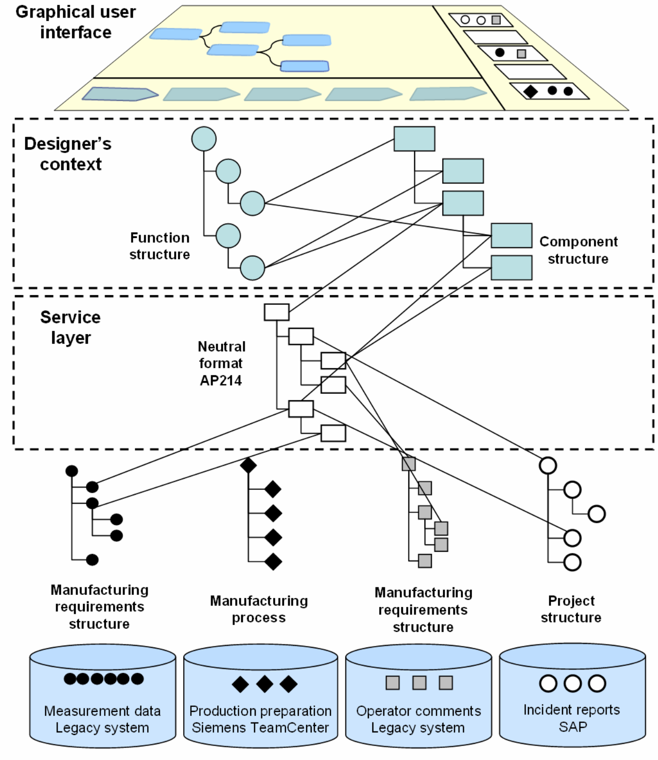


Figure 9 – Demonstrator architecture

needed to access and gather information on one hand and define the context in which the accessed information will be logical to the designer on the other. The different steps were needed due to the fact that in some databases data is structured according to the structure of manufacturing requirements, in some according to the manufacturing process and in some according to different projects. But what the designer wants to see is the data structured according to the function structure and component structure.

The access to and integration of the four different information sources will be enabled by a service oriented architecture. A similar approach has been reported by Chen et. al. [18] where a typical collaboration manufacturing model for virtual manufacturing enterprise alliance is presented. For the the SOA implementation the standard PLM Services 2.0 is considered in order to evaluate the standard and also enable the desired demonstrator characteristics. This approach has been chosen in order to enable the flexibility to expand the scope of the demonstrator and to also enable for other existing or new applications/portals to access the information that is made available through this integration. The flexibility also enables the integration of more information sources.

The presentation of the accessed and reformatted information is done by a client application with a specific graphical user interface (GUI). The client application contains the function structure and component structure to which information from the information sources is linked and presented.

A screenshot of the GUI is shown in Figure 10. In the main area there is the ability to switch between the component structure and function structure. There is also an ability to apply a project filter in order to only show manufacturing experience related to a specific development project.

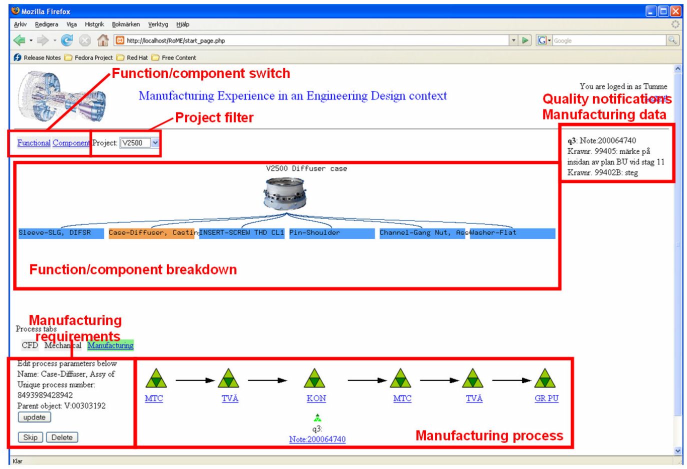


Figure 10 – Graphical User Interface of the demonstrator

In the process area the manufacturing process for the selected component or function is presented. The process is stored in Siemens TeamCenter. In the area where quality notifications and manufacturing data is presented operator comments, stored in SAP/R3, and data from manufacturing measurements, stored in a legacy database, will be presented for the selected component or function. The same is done in the area showing manufacturing requirements which are stored in SAP/R3. The amount of results in these two fields can be narrowed down further by choosing a specific manufacturing activity.

The workflow for the demonstrator is that the designer chooses the function/component and/or project whose manufacturing experience he/she is interested in. The process field is automatically updated showing the manufacturing process for that particular component/function. Quality notifications, manufacturing data and manufacturing requirements for that particular component/function are automatically updated. If the designer is interested in manufacturing experience related to a specific step in the process, e.g. welding, the requirements, manufacturing data and quality notifications are updated so that they now only show information relevant for the chosen component/function, project and the welding step of the process.

The layout of the GUI along with the fact that information will be dynamically accessed and presented as the user selects components or functions creates the context in which the manufacturing data becomes more logical from a designer's point of view.

8 DISCUSSION AND FUTURE WORK

The focus of this paper has been to describe a solution for the re-use of manufacturing experience in early lifecycle phases in order to make the product easier and faster to produce. The general and more abstract idea is that information gathered in a later lifecycle phase is fed back to earlier phases in order to be able to optimize the product over a larger portion of the lifecycle. The described concept can be extended to include all lifecycle phases so that the optimization of the product can extend over the whole lifecycle thus enabling the realization of product lifecycle management to a greater extent. The experience can be in the form of documents such as design guidelines but it can also be documented in the form of video clips or online demo presentations such as those exemplified at Honeywell [28].

Using the process, together with the product, as a means for structuring different kinds of experiences has been found to be feasible and will be further evaluated. By applying the process perspective experience from e.g. calculations performed during the development or service actions performed during the aftermarket of a component or even an individual of a component can be made accessible in an easy way. Developing the proposed GUI to entail also other processes in the product lifecycle and make experience from those lifecycle phases available will not be a large task due to the generality of the proposed GUI structure. The access to the information sources containing the experience will be secured by connecting those sources to the information service layer.

The future work entails the development of the information service layer described in Sections 4 and 5. The developed demonstrator, once the service layer is implemented, will be expanded by another way of contextualizing the manufacturing experience. The information will be made available even closer to the designer and the designer's context by connecting a CAD integrated KBE application which will use the manufacturing experience in order to optimize component from a design for manufacturing perspective and be able to take into account the latest information from the manufacturing system.

9 CONCLUSIONS

This paper concludes that contextualization of and the ability to access manufacturing data in real time are two key enablers for providing design engineers with manufacturing experience from earlier and ongoing projects. The approach to visualize data from dispersed sources in manufacturing using web technology and with a design engineer's perspective provides a powerful engineering tool in the early phases of product development. The service oriented PLM architecture enables access of manufacturing experience in a dispersed system environment and provides the possibility to integrate knowledge based engineering applications which focus on DFM in order to provide them with real time input data from manufacturing.

10 REFERENCES

[1] Andersson, P., Isaksson, O., 2008, “Manufacturing system to support design concept and reuse of manufacturing experience”, draft paper submitted to 41st CIRP Conference on Manufacturing Systems, Tokyo, Japan

[2] Andersson, P., Wolgast, A., Isaksson, O., 2008, “Current industrial practices for re-use of manufacturing experience in a multidisciplinary design perspective”, draft paper submitted to DESIGN 2008, Dubrovnik, Croatia

[3] Swift, K. G., Booker, J. D., 2003, “Process selection: from design to manufacture”, Butterworth, Oxford, UK

[4] Boart, P., Isaksson, I., 2006, “Enabling variation of manufacturing process parameters in early stages of product development”, ASME, IMECE2006-14459.

[5] Boart, P., Nergård, H., Sandberg, M., Larsson, T., 2005, “A multidisciplinary design tool with downstream processes embedded for conceptual design and evaluation”, ICED 05.

[6] Boart, P., Andersson, A. and Elfström, B-O., 2006, “Knowledge Enabled Pre-processing for Structural Analysis”, 1st Nordic Conference on Product Lifecycle Management - NordPLM'06.

[7] Sandberg, M., Boart, P., Larsson, T., 2005, “Functional Product Life-cycle Simulation Model for Cost Estimation in Conceptual Design of Jet Engine Components”, Journal of Concurrent Engineering, Vol:13/4 pp331-342.

[8] Åström, Peter - Doctoral thesis: “Simulation of Manufacturing Processes in Product Development” - 2004:56 • ISSN: 1402 -1544 • ISRN: LTU - DT -- 04⁄56 -- SE

[9] Molina, A., Aca, J. and Wright, P. (2005) 'Global collaborative engineering environment for integrated product development', International Journal of Computer Integrated Manufacturing, 18:8, 635 - 651

[10] Alizon, F., Shooter, S. B., Simpson, T. W., 2006, “Reuse of Manufacturing Knowledge to Facilitate Platform-Based Product Realization”, ASME, Vol:6 pp170-178

[11] Burr, H., Vielhaber, M., Deubel, T., Weber, C., Haasis, S., 2005, “CAx/Engineering Data Management Integration: Enabler for Methodical Benefits in the Design Process”, Journal of Engieering Design, Vol. 16, pp 385-398

[12] Bergsjö, D., Burr, H., Malvius, D., Müller, M., Vielhaber, M., 2007, ”Product Lifecycle Management for Cross-X Engineering Design”, Proceedings of ICED 07, Paris, France

[13] www.omg.org

[14] Bergsjö, D., Ćatić, A., Malmqvist, J., 2008, “Implementing a Service Oriented PLM Architecture using PLM Services 2.0”, draft paper submitted to DESIGN 2008, Dubrovnik, Croatia

[15] Shaffer, J., Kopena, J. B., Regli W. C., 2007, “Web Service Interfaces for Design Repositories”, Proceedings of DETC2007, Las Vegas, Nevada, USA

[16] Ćatić, A., Malmqvist, J., 2007, “Towards Integration of KBE and PLM”, Proceedings of ICED 07, Paris, France

[17] Yang, Q. Z., Lu, W. F., 2007, ”A Web-Enabled Engineering Object Modeling Environment to Support Interoperability and Intelligent Services in Collaborative Design”, Proceedings of DETC2007, Las Vegas, Nevada, USA

[18] Chen, Q., Shen, J., Dong, Y., Dai, J., Xu, W., 2006, “Building a Collaborative Manufacturing System on an Extensible SOA Based Platform”, Computer Supported Collaborative Work in Design, DOI: 10.1109/CSCWD.2006.253179

[19] Srinivasan, V., Lämmer, L., Vettermann, S., 2008, ”On Architecting and Implementing a Product Information

Sharing Service", Journal of Computing and Information Science in Engineering, vol. 8

[20] Abramovici, M., Bellalouna, F., 2008, "Service Oriented Architecture for the Integration of Domain-Specific PLM Systems Within the Mechatronic Product Development", Proceedings of TMCE 2008 Symposium, Izmir, Turkey

[21] www.oasis-open.org

[22] www.vivaceproject.com

[23] www.vec-hub.org

[24] Bazire, M., Brézillon, P., 2005, "Understanding Context Before Using It", CONTEXT 2005, pp. 29-40

[25] Dey, A.K, Abowd, G.D., 1999, "Toward a Better Understanding of Context and Context Awareness", Handheld and Ubiquitous Computing, v. 1707, pp. 304-307

[26] Redon, R., Larsson., A., Leblond, R., Longueville, B., 2007, "VIVACE Context Based Search Platform", CONTEXT 2007, pp. 397-410

[27] Andreasen, M. M., 1992, "The Theory of Domains", Proceedings Workshop Understanding Function and Function-to-form Evolution, Cambridge, UK

[28] Thilmany, J., 2004, "Share the Wealth", Mechanical Engineering, v. 126, no. 12, pp. 32-35

Paper D

Requirements Management When Introducing New Mechatronic Sub-systems – Managing the Knowledge Gaps

Ćatić, A. and Malmqvist J. (2010) Proceedings of 11th International Design Conference DESIGN2010, Dubrovnik, Croatia, pp 661-672

INTERNATIONAL DESIGN CONFERENCE - DESIGN 2010
Dubrovnik - Croatia, May 17 - 20, 2010.

REQUIREMENTS MANAGEMENT WHEN INTRODUCING NEW MECHATRONIC SUB-SYSTEMS – MANAGING THE KNOWLEDGE GAPS

A. Ćatić, J. Malmqvist

Keywords: Mechatronics, requirements management, knowledge gap

1. Introduction

The "mechatronization" of drivelines with the aim to reduce fuel consumption and emissions is occurring across the commercial vehicle industry as a response to rising environmental awareness and fuel prices. This has given rise to new challenges in the commercial vehicle business processes. Commercial vehicles are business-to-business products. The requirements governing their designs have been heavily focused on performance in terms of power, carrying capacity and low maintenance cost which are the main factors affecting the lifecycle cost for the customer. These circumstances have resulted in the design efforts and resources traditionally being focused on mechanical systems and properties. The new challenges lie in the needs to manage a rapidly increasing number of functions along with different processes and a different culture that accompanies development of electronics and embedded software.

The research about development of mechatronic products is ranging from methods and processes to specific IT support and IT architectures. Many of the issues found are general product development issues for which there are recommendations and guidelines proposed in product development literature such as [Ullman, 1997] or [Ulrich and Eppinger, 2008]. Focusing on the specifics of mechatronic product development, a key issue is to ensure the most effective integration of the three involved domains of mechanics, electronics and software. Many research efforts are directed towards developing new methods and adapting existing methods to address the issues which arise in the integration of the three domains e.g. focusing on cross-domain interface and requirements management, roles and responsibilities, process and information management and verification and validation management.

[Almefelt et al, 2006] have studied an industrial case and from this empirically derived a set of recommendations for requirements management which, among other things, address the need to early define and focus a certain set of over-arching cross-system requirements. They also address the need to clarify each requirement with the underlying context and intent and define interfaces and verification methods for each requirement. Other contributions such as e.g. [Jansen and Welp, 2007] suggest models to primarily overcome the interface related issues by identifying and classifying different kinds of interfaces.

[Adamsson, 2007] addresses managerial implications of mechatronic product development and presents a set of proposals regarding increased awareness of the importance of the embedded software as well as the need to organize for cross-domain collaboration together with a reconsideration of the recruitment strategy in order to make sure the competencies of the project participants reflect the three domains present in the product. Other recommendations from [Adamsson, 2007] address the need to communicate a clear cross-domain integration strategy and the need to communicate that the product launch is not only about manufacturability but also validation of embedded software.

[Bergsjö, 2009] considers process and information management issues and identifies from empirical studies that the different domains have different processes with different information needs and IT tools and presents different ways of integrating the process and IT environments.

As a response to the need for specific methods and processes a VDI guideline called VDI2206 [VDI2206, 2004] has been developed which is based on the systems engineering methodology and addresses requirements and interface management issues along with verification and validation pointing towards useable IT tools for modelling, and simulation. The guideline also gives a rough overview of how a mechatronic product is matured from an initial idea to production readiness. The research community has tested the method in several cases [Bathelt et. al., 2005] [Rahmnan et. al., 2007] [Ziemniak et. al., 2009].

The literature in the field of development of mechatronic products thus covers many different areas and aspects as described above. However, the earlier contributions have been focusing on managing embedded software in different ways [Adamsson, 2007][Bergsjö, 2009] or on applying methods for developing mechatronic products [Bathelt et. al., 2005] [Rahmnan et. al., 2007] [Ziemniak et. al., 2009]. The gap identified in the current research regards the management of the new situation where OEMs have to integrate systems, rather than components, and assure overall functions, manage interfaces and harmonize supplied systems with each other.

The aim of this paper is to focus on interface related issues in development of products containing new, and largely supplied, mechatronic sub-systems. The purpose is to, based on an empirical study, produce a set of recommendations which focus on managing knowledge gaps as a way to manage the new situation and complement other recommendations and guidelines present in the literature. The research question driving this effort has been:

Which issues arise when a new and supplied mechatronic sub-system is integrated into an electronically controlled mechanical product? How can these issues be managed?

2. Empirical setting and research method

The research study was initiated with a wish to study how knowledge gaps were identified and knowledge reused in a project whose main goal was to develop a new driveline containing a new mechatronic sub-system which adds a substantial amount of new functions, interfaces and suppliers.

The study was initiated with two workshops where the line manager of the new department, responsible for the new sub-system, and the chief project manager for the development project were consulted for issues which they wanted to investigate closer. The result from these workshops was a list of issues which were frequent during the development project and needed to be focused and clarified. The focused issues were requirements management, system interfaces, the limitations of the present component oriented line organization, and supplier management.

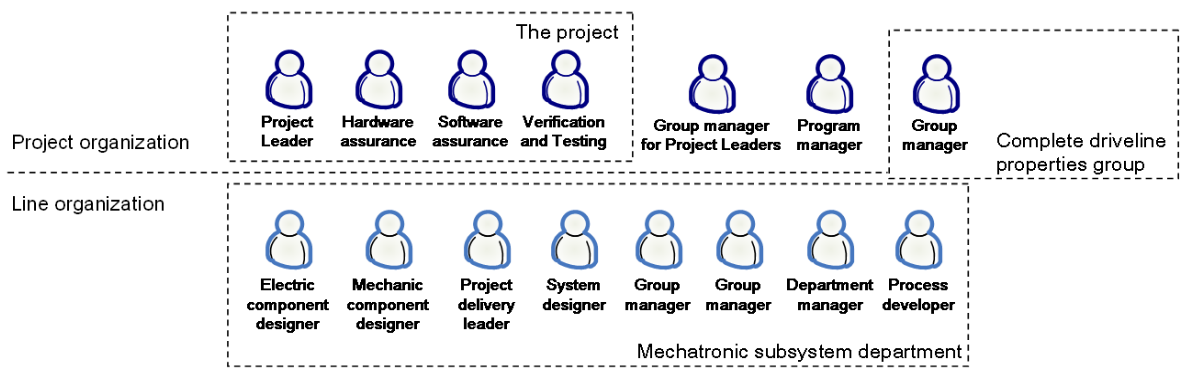


Figure 1 – The interviewees placed according to their relations in the matrix organization

A total of 15 interviewees were chosen from both the new line department and the project team, see Figure 1. These were interviewed in semi-structured interviews and were asked to reflect around the focused issues. The interviews were followed by a literature review within the focused issues and a set of prescriptive elements such as recommendations were found. These were later discussed in a set of workshops with chosen interviewees. The elements found were related to supplier management and integration in new product development by [Johnsen, 2009] and [Ragatz et. al., 1997], requirements management [Hull et. al., 2005] and [Almefelt et. al., 2006] and knowledge management methods from the lean product development movement [Kennedy et. al., 2008]. Findings regarding interfaces have been related to modularization and platform design with methods on how to define and describe interfaces [Ulrich and Eppinger, 2008] but not so much on how to document and manage them.

3. Presentation of findings

In this section, the empirical findings regarding requirements management and verification, interface management, supplier management and the limitations of the present component oriented line organization from the interviews are described in more detail.

3.1. Requirements management process

The first high level requirements originate between product planning and the customer with the purpose to frame the scope for the project and are quite general. These are then handed over to the project that makes additions necessary to be able to forward these to the line departments. An issue of critical importance from the line department's point of view is the definition and allocation of requirements to each sub-system of the driveline (e.g. transmission, base engine etc.) because this affects the ability to define and allocate requirements to each component within the sub-system. In the case of the new line department there was a problem to derive detailed requirements needed at the component level due to lack of a legacy and previous knowledge of the mechatronic sub-system both among project and line members, see quotes in Figure 2. This was solved by an iterative approach where the component designer made a qualified guess and went back with a solution onto which the project members could react and refine the requirements. At best this approach was time consuming and at worst it was both time consuming and frustrating. The flow of requirements is top-down driven and there is little preparedness for managing requirements which go the other way e.g. that a certain choice of material on one component restricts the temperature emitted by another component. What usually happens is that these issues are discovered during testing of prototypes and lead to late and expensive changes. During the interviews several of the interviewees mentioned similar problems from an earlier project with a new, but significantly smaller, mechatronical sub-system. For the other (predominantly mechanical) sub-systems the component interfaces are known and such issues considered early on.

3.2. Requirements management methods and tools

Every forum that carries a responsibility of requirements definition and allocation keeps their requirements lists in their own Microsoft Excel files, as illustrated in Figure 2. At the component level this means that there is a specification per component stored locally in each component designer's computer. Updates across specification documents are time consuming and error prone, negatively affecting management of cross-component and cross-sub-system requirements. The top-down driven flow of requirements is also reflected in the methods and IT tools which provide no support for a backflow of requirements. Interviewees claimed that this impairs both the creativity of the designer and the efficiency of the project. If a component designer is able to propose different solutions and say that depending on which one is chosen different requirements are posed back on other solutions in the system, system related issues could have been detected earlier. The assumption in the top-down oriented requirements management process is that requirements can be clearly allocated to each component and that the system of components will function optimally. Since this is not the case the

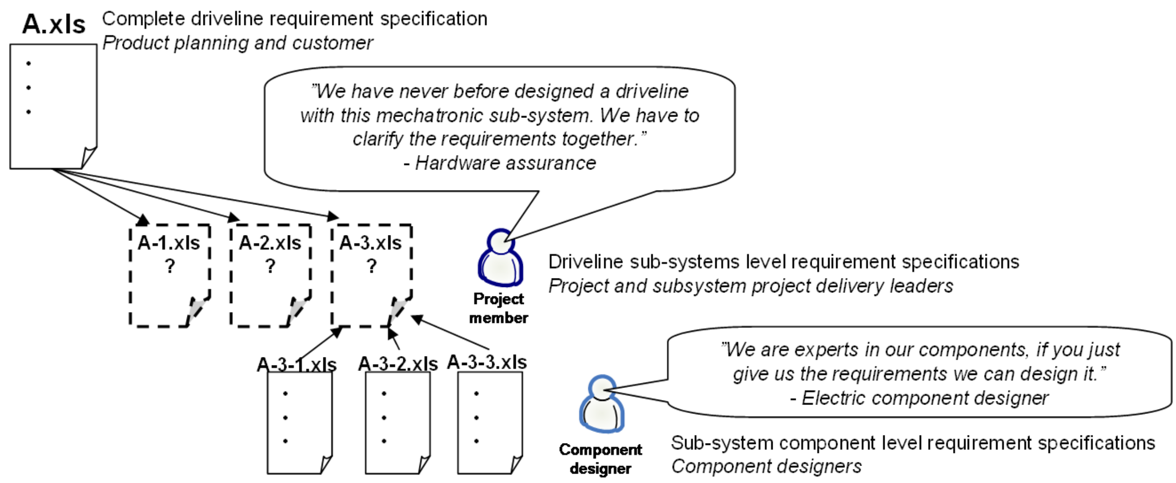


Figure 2 – Lack of sub-system level requirement specifications and interviewee quotes.

complete approach in was based on testing and prototypes to detect system related issues and discover unknown interfaces.

3.3. Requirements formulation and prioritization

Much focus regarding the requirements themselves was directed to their formulation. Issues like fuzziness in the way the requirements are formulated have been stated by the interviewees. Some of the members of the line department suggested that quantitative requirements should be formulated in intervals because this would enhance the possibility to consider several solutions and be more creative.

Another issue related to the formulation of the requirements is the lack of background and context for each requirement. Many of the respondents, both from the project and the line department, were positive to providing more context in the form of background, change history (with reasons for change) and rationale for specific requirements, which is hard to do in Microsoft Excel. Relations between requirements were also unclear both in terms of how requirements relate to each other in the top-down flow but also how different requirements at the same level relate to each other e.g. across components or sub-systems, which is once again due to the file based requirements management in different Excel sheets, see Figure 2. Two of the respondents argued that an increased context for the requirements would also increase the ability to innovate because a context would provide the possibility to understand the general idea and consider new combinations of solutions.

Regarding mechanical, electrical and software requirements some of the respondents have stated that the designer's background has affected which requirements were prioritized, exemplifying with a mechanical engineer who tended to focus on mechanical requirements.

3.4. Lack of requirements legacy and knowledge

During the interviews several of the respondents referred to the issue of "missing requirements", which represented different phenomena depending on the background of the respondent. When respondents from the line department talked about missing requirements they referred to lacking technical knowledge among the project members in how to define and allocate requirements, resulting in "missing requirements". An example is that the component designer for an electric motor expected specified torques and boundary conditions from the project members who simply could not derive such detailed information from the high level requirements coming from product planning such as "reduce fuel consumption by X percent". When the project members referred to "missing requirements" they see it as a natural consequence of the fact that there is no legacy from which to carry over requirements. Their perception, on the other hand, is that there is a lacking individual responsibility among the component designers to drive and, starting from the fuzzy high level requirements, gain knowledge and define requirements for their components. This misunderstanding of whose responsibility it is to define sub-system level requirements is illustrated in Figure 2.

3.5. Verification management

In the process of defining and performing the verification of requirements the interviewees answered that the verification methods are defined very late in the process and it is common practice that this was done by the component designers alone which was perceived as strange by the interviewees because they did not ensure that the chosen verification methods were congruent with existing methods and equipment in the test department. They were sometimes not even informed that certain equipment was needed.

The interviewees requested more methods and tools for early verification in order to guide the project to a quicker convergence on the final requirements and on conceptual solutions that are to be used. Suggestions on what this early verification might be included more simulation and more physical test benches for sub-systems.

Finally the issue of verification of supplier components as opposed to in-house components was stated as needing more attention. Several of the respondents argued that verification planning should consider the fact that if a failure is discovered on a supplied component there is a much longer loop of reporting and redesign before the component can be tested again than if an in-house component fails.

3.6. Supplier management and management of concealed requirements

In the particular case of the observed mechatronic sub-system an apparent issue that the component designers had to deal with was the suppliers' knowledge gaps regarding technical automotive standards like sealing, vibrations, temperatures and so on. This was claimed to be the most concerning issue that affected the work.

Several respondents requested that a supplier management process be set up which would be in harmony with the stage-gate process that governs the development project in order to be able to harmonize the work and deliveries from the supplier with the rest of the project deliveries.

Another specific issue for this particular project is that certain critical components only had one supplier. This made it critical to manage the relationship with that supplier in a good way in order not to jeopardize the complete project. For one particular component the requirement specification was perceived by the supplier as too tough. Even the component designer realized this in retrospect and the reflection was that the uncertainty in the complete project caused him to set requirements with a large safety margin out of precaution. The requirements however almost caused the supplier to terminate the partnership and the result was that the requirements specification was revised and a set of follow-up requirements were communicated back from the supplier which was hard to manage in the top-down driven flow of requirements described earlier.

"Management of concealed requirements" was mentioned by one of the respondents and is considered mainly as a consequence of the suppliers' lack of experience of the automotive industry. These requirements are sometimes not even explicitly stated because they are considered as "industry standard". Initially this designer's attitude towards the supplier was that even those requirements which are obvious from an automotive point of view should be stated. However it turned out that due to the fact that these requirements were explicitly stated the price tag from the suppliers was significantly increased. What he noticed was that the component fulfilled these requirements even when they were not explicitly stated which meant that they could be taken out of the contract but they would still have to be verified in the internal verification processes. This was referred to as "management of concealed requirements".

3.7. Interface management in the component oriented line organization

The component oriented line organization is set up with the top-down driven flow of requirements in mind. It is also set up with the assumption that all interfaces are known and well-defined in order to facilitate the allocation of requirements. As the project has had a high level of new components with

unknown interfaces a top-down driven flow of requirements was not possible and it resulted in certain requirements falling between component designers only to create a chaos later in the project. The organization has tried to compensate for this as much as possible by arranging "interface meetings" as soon as critical interface issues were discovered. These meetings generated a slowness and a frustration in the project. If two component designers from completely different sub-system departments found an issue that neither of them felt responsible for the project leader was the only higher instance with the mandate to decide, which created frustration.
Several interviewees proposed a function oriented organization, with function owners, to balance the component focus, in order to manage the increased complexity. Two of the interviewees had been working in another automotive company where a functional organization complements the component organization. One of them was very positive and had several successful examples while the other had a negative experience of this organization mainly related to requirements management. His perception was that the two different dimensions led to a behaviour of "over-specification". Function and component specifications overlapped and even conflicted but it was impossible for a single person to be able to detect this due to the extensiveness of the specification documents (> 2000 pages).
Among the other interviewees the notion of function owners was perceived as positive. Some even considered this as a prerequisite for being able to deliver drivelines on time and quality and with an ever increasing content of mechatronic subsystems. Two questions were however raised:

1. Organization (will existing component or systems designers have a partial role as function owners or will function owners be completely new people?)
2. Mandate (which responsibilities should function owners have in relation to the project leader and the line managers?).

One respondent claimed that introducing function owners risks to create a bigger mess. Instead a clarification and accentuation of personal responsibility for driving undefined issues such as detailing of requirements and exploration and management of interfaces.

4. Analysis of findings

This section summarizes and discusses the findings from Section 3 using the notion of "knowledge gaps", as defined by [Kennedy et. al., 2008] and summarized in Section 4.1, in two different categories, requirements management practice and supplier management. The reason why most of the findings and the analysis are about requirements management is that requirements are the main driver of the product development project and it is through issues found in the requirements management process that issues related to interface management and suppliers management are revealed.

4.1. Knowledge gap – the definition

In the Lean Product development paradigm, as described by [Kennedy et. al., 2008], one of the key terms is labelled the knowledge gap. In order to understand the knowledge gap product development is divided into two value streams, as illustrated in Figure 3. The product value stream is what is traditionally labelled as the "Product Development Process" in most companies. The product development process is usually gated with stages and deliverables and is believed to cover all aspects

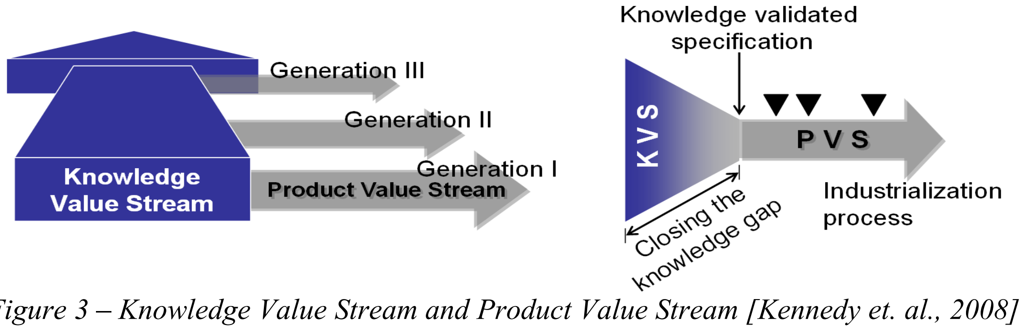


Figure 3 – Knowledge Value Stream and Product Value Stream [Kennedy et. al., 2008]

of what is needed to be managed. In these processes knowledge is more or less systematically transferred between projects but is not explicitly viewed as a deliverable, the product design is the major deliverable. Even in the academic models of product development processes by e.g. [Ullman, 1997] or [Ulrich and Eppinger, 2008] the product design is the major deliverable. The major difference between traditional and lean product development is thus that knowledge is defined as an explicit deliverable in the latter and is given a value stream of its' own. The knowledge value stream (KVS) flows across projects and there are methods for managing it in and between projects. This is the major reason why this is considered as "lean", since having a strategy and methods for capturing and reusing knowledge lowers the risk of repeating mistakes, thus making the product value stream (PVS) more efficient and more effective.
According to [Kennedy et. al., 2008] knowledge gaps may arise in the interface between the KVS and PVS. The question that is posed is simply "what is known" and "what is needed to be known in order to reduce the risks prior to initiating the PVS for a certain product". The answers to these questions assume that the company is aware of which parts of the product solution have the highest uncertainty. This uncertainty needs to be eliminated before the PVS starts. The closing of the knowledge gap is characterized by analytical activities to see in which intervals of requirements different principal solutions work and by doing so see which are viable for the current projects requirements.

4.2. Requirements management practice

The requirements management process in the studied case is adapted to the company's standard scenario: a project which develops a driveline with the same architecture in terms of included sub-systems and components as previous drivelines. This standard scenario presumes that:

1. Most components are mechanical, a few electrical and there is a central control system containing almost all the software elements in the driveline.
2. There is a well-known legacy for each sub-system and for most components in the form of previous requirements, solutions, test results and suppliers which can be used as a base line.

The effects of the first presumption are that a system is divided into components for which component designers can be assigned. For each component, its' boundaries will give a clear enough view of which interfaces that need to be taken care of. For the owners this translates into which people they need to keep in touch with. The relatively low number of electrical components, all of which have quite limited number of functions, makes it possible for a few designers to keep track of them. The concentration of all the software to one group of designers responsible for "control systems", makes it a concentrated function which can be isolated in the development process and integrated only at major releases. The fact that all of the software is developed in-house does not necessitate any detailed processes in how revisions are done since most of the issues discovered during testing can be fixed more or less simultaneously. This set up is depicted in the left part of Figure 4.

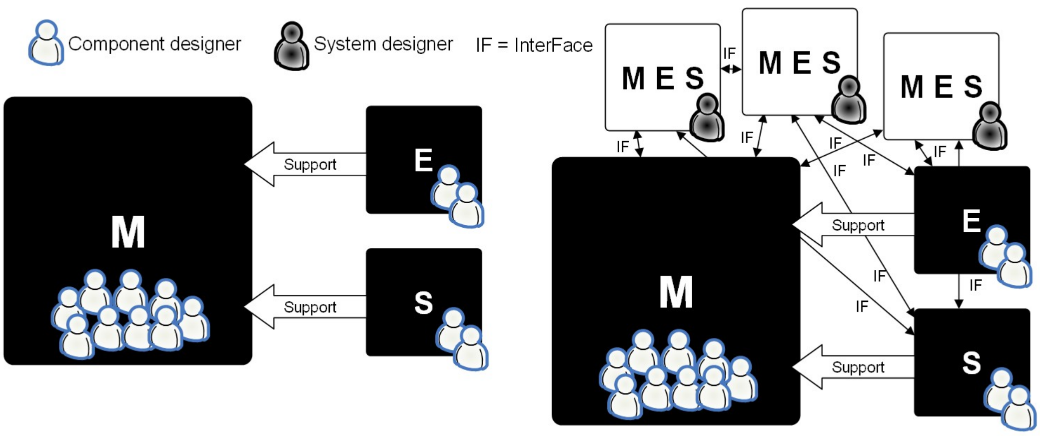


Figure 4 – Left: electronically controlled mechanical system.
Right: Addition of the mechatronical (MES) sub-system.

The second presumption is that knowledge gaps between two projects are small enough to be managed by the individual component designers themselves and do not pose any big risks. The set up depicted in the right part of Figure 4 resembles the scenario in the studied project. The characteristics of this set up are:

1. Most components are still mechanical but a substantial amount of the drivelines functions relies on components which contain both mechanical, electrical and software parts. Software functions are thus scattered around in different parts of the system making the interactions in the structure of the driveline much more complex.
2. Due to the novelty of the new mechatronical sub-system there is no legacy for the new components.

Since the company processes and organization are set up according to the situation in the left part of Figure 4 each of the new MES sub-systems had only one component designer and this person's background affected which of the three domains that were focused in the requirement specification towards the supplier. In addition, this one person had to deal with many unknown interfaces in the three domains. The lack of legacy caused a large knowledge gap in both the interfaces and the requirements which caused schedule slips and late changes. The interviewee who had the role of system designer was the one responsible of figuring out which interfaces there are in the three domains and could witness of a piling amount of testing failures which all were results of missed requirements due to unknown interfaces.

In addition to these knowledge gaps, the group manager for complete driveline properties said that the requirements breakdown process was terminated as they were "in the middle of the left part of the systems engineering V" when detailed design started. A void was thus formed between what kind of requirements the component designers needed and what kind of requirements that were coming from the project. This void was further maintained by the fact that the component designers were expecting the project to concretize the requirements onto a level they could use and the project expected the component designers to take an initiative and do the concretization (as depicted in the quotes in Figure 2). According to literature on requirements management [Almefelt et. al., 2006] and [Hull, 2005] such a void is bound to give rise to gaps and overlaps between sub-systems and components due to the loss of traceability between system requirements and component requirements. This is also what happened in some interface meetings when requirements "fell between chairs", as some of the designers put it.

The issues of requirements formulation with lacking context and lacking verification criteria are a consequence of the large knowledge gap initially in the project. These issues are in no way unique for this project and are quite general for any project in the studied company. However, when the component designers were asked about how they dealt with those requirements which belonged to the group "official or unofficial automotive standards" regarding e.g. vibration and sealing, they said that they could get some background and context by talking to experienced designers. This implies that when legacy requirements lack background and context these still exist implicitly in some experienced designer's head (meaning that the knowledge gap for those requirements is not as big as it seems). When context and background are lacking for requirements related to the new sub-system this knowledge can not be found anywhere in the organization, thus the knowledge gap is large. Therefore it is doubtful that, even if the project got more time as requested by the complete driveline properties group manager, they would be able to derive a complete set of sub-system and component requirements simply because nobody had the knowledge enough to do that. A way to strategically address the issue of a knowledge gap according to [Kennedy et. al., 2008] is to first make all of the involved parties aware of the existence of such a knowledge gap and actively work on reducing it by modelling and simulating or even testing specific characteristics to generate as much knowledge about the behaviour of chosen parts of the system as possible. The key, according to [Kennedy et. al., 2008], is to build small and simple rather than detailed and all-embracing models and tests but big enough to gain the specific knowledge and close the gap. This need for early testing and simulation was expressed also by the interviewees in the discussions around verification management in Section 3.6.

This would have given valuable input to the component designers both on how to specify requirements and verifications and manage interfaces.

4.3. Supplier management

In a study by [Johnsen, 2009] three decades of research regarding supplier involvement in new product development and innovation is summarized. The findings from this extensive research review are summarized by [Johnsen, 2009] in Figure 5.

In the first category, supplier selection, the first factor is early supplier involvement which means involvement during the concept stage or during early feasibility studies. The question is whether this factor has been considered or not in the studied case. The development of the system architecture, that was set already by the internal R&D division in early phases, seems not to have involved any suppliers. This would mean that the suppliers were involved later during embodiment design and detailed design. There are no answers that indicate that the factors of supplier roles and involvement and innovative capability of the suppliers have been given any focus. An issue regarding supplier selection that is not discussed by [Johnsen, 2009] is that new product development involves new technology and that the amount of suppliers may be very limited which has been the case for most components in the mechatronic sub-system.

The second category states several factors which have been identified also by [Ragatz et. al., 1997] who in their study statistically evaluated different management practices and environment factors which distinguished successful and unsuccessful supplier integration in new product development. They point out that factors related to social, legal and organizational aspects were far more important than aspects related to technological difficulty and complexity. They emphasize that managing the relationship with the supplier through common trust, common goals and visions and involvement has a greater impact than managing them through formal processes and documents. It can be said that many of these factors, such as common trust and goals are missing in the studied project. The interviewees even considered that a higher grade of formality towards suppliers would have been beneficial to the project. Regarding the factor of agreed performance targets and measures in the second category it has been highlighted by the interviewees that they would like to see a higher involvement of the suppliers in the validation of requirements and in verification and testing in order to shorten subsequent redesign loops.

Many saw the low involvement as a result of the lacking structure in validation and verification processes at the OEM. In the third category of internal customer capabilities the second factor is described as the ability to manage cross-functional relationships in order to manage supplier relationships. This correlates well with the found interface issues in the component organization that also affected the integration of suppliers negatively. The recommendation by [Kennedy et. al., 2008] to map out the knowledge gaps of the involved parties early on in the project is also valid for the

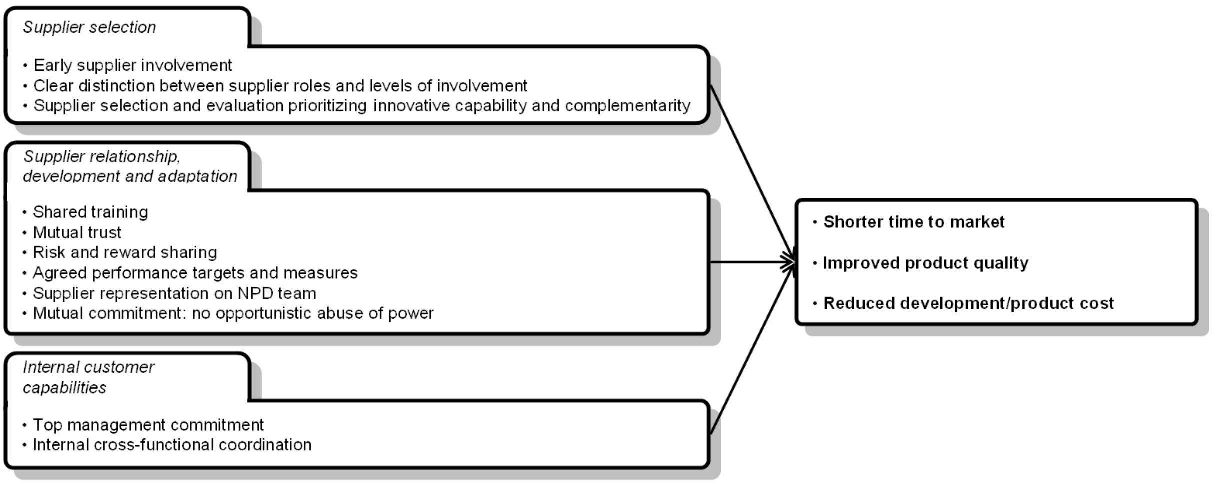


Figure 5 – Factors affecting supplier integration in new product development [Johnsen 2009]

suppliers since it was indicated by the interviewees that there were substantial knowledge gaps regarding automotive standards and OEMs processes. These gaps were not mapped out as potential risks and dealt with openly with the suppliers which would be in harmony with the recommendation by [Johnsen, 2009] to manage this relationship based on common trust, goals, visions and involvement rather than formalized contracts. Considering for example the problems with the long software redesign loops for supplied components, compared to short redesign loops for internally developed software, a higher degree of formality in the relationship to the supplier would tend to slow down the redesign process.

4.4. Knowledge gap in the studied case

The studied project was started in the same way as a project with mostly mechanical parts and no new sub-systems. One of the group managers put this very concretely in the following statement:

> *"All sub-systems got equal attention and resource allocation which is strange given the fact that issues in the transmission are more explored and well known than those in the our sub-system."*

Project planning assured the effectiveness of the project and provided a set of high level requirements. The knowledge gap was implicitly assumed to have been closed by means of experienced designers in the roles of component designers in the mechatronic sub-system. As the sub-system architecture was delivered by the R&D division this was considered as equal to regular carry-over for existing sub-systems. The legacy implications described earlier however show that the amount of carry-over in terms of legacy in requirements, supplier abilities and interfaces had a larger knowledge gap than expected. What was especially lacking were:

1. Functional as well as safety requirements regarding the embedded control software
2. Software interfaces which result from the embedded ECUs in the new components
3. Physical interfaces towards other sub-system components which become apparent only when the solution is tightly packed in the vehicle
4. Interfaces towards equipment in manufacturing and after market
5. Knowledge about supplier abilities and supplier processes for e.g. redesign tasks

Another knowledge gap was found regarding the process needs for the design of the new mechatronic sub-system in relation to characteristics of the existing processes. This relates back to what was mentioned in how system related issues are detected. The existing process fits very well the needs of a development project in which no new sub-systems are added to the driveline. The basic set up of the process is that the designers already know which interfaces their components have and most system effects are known. The complete driveline tests are there only to ensure that any unexpected system behaviour is resolved before production. This approach is however not suitable when a new sub-system with above stated knowledge gaps is introduced since there are too many unexpected system effects. This was not realized by those involved in the project and thus was a knowledge gap in itself.

5. Conclusions and recommendations

This empirical study has found a set of different problem areas related to the introduction of a new mechatronical sub-system to an existing and mature product, a commercial vehicle driveline. Based on the findings and the sub-sequent analysis it is clear that the root cause of the problems encountered in the product development project is the lack of legacy for the new sub-system. The development processes, methods and IT tools at the studied company presuppose that each sub-system already has a base of knowledge which can be carried over regarding sub-system and component interfaces, requirements, verification methods and suppliers. From the presented findings it is clear that substantial knowledge gaps existed for each of these categories for the new mechatronical sub-system. This constitutes an answer to the "which issues" part of the research question stated in the beginning.

The second part of the question addresses the issues of integration of new mechatronical sub-system. This part of the question was stated due to the trends which are going on across the complete automotive industry and the fact that automotive companies are increasingly becoming system integrators rather than component integrators, as illustrated in Figure 5. The component designers are however mainly knowledgeable in only one domain which, combined with the lack of legacy, means that there are substantial knowledge gaps which may be hard for the component designer to realize.

The fact that a new sub-system may be completely supplied by suppliers with no automotive experience makes the consequences of the above stated issues much deeper. As stated in the findings it is hard enough to use the existing processes, methods and IT tools to manage requirements and coordinate the component designers for the new sub-system. Adding the fact that these also have to interplay with the processes of the suppliers creates a situation out of control.

Based on these conclusions the recommendations in Table 1 are proposed and constitute the prescriptive part of this study. A comment on the recommendations is that due to the fact that the delivery from the R&D division was assumed to constitute a regular carry-over the recommendations are geared towards the R&D process, the transfer process and the product development project initiation process. The references in the table are those with similar or same recommendations.

Table 1 – Recommendations when transferring and introducing a new mechatronical sub-system into an electronically controlled mechanical system

Issues found	**Recommendations**
Unknown interfaces in the three domains.	Knowledge about interfaces of the new sub-system components towards other sub-systems and components and how these are dealt with in all the domains needs to be built up or knowledge gaps visualized prior to implementation [Nobelius, 2002][Almefelt et. al., 2006]. Functional and geometrical interfaces should be explored and carefully documented but also other interfaces, e.g. electromagnetic, thermal or vibrational interactions between components which result when the solution is packed into a vehicle.
Lack of processes for software development and verification.	There should be a “software developing culture” and acceptance of such roles in the implementing organization and an awareness of the increased functional content in the completely new system [Adamsson, 2007].
Unknown interfaces in the software domain.	The responsibility for functions which rely on software should be clear and agreed upon between the internal roles of software developers in the central control system and the component designers who are responsible for a component or small sub-system that carries an embedded ECU and software.
Lack of processes, methods and IT tools for requirements management and verification.	Processes, methods and IT tools needed for the detailed design and adapted to the new issues found in the new sub-system which the existing sub-systems have not encountered. The existing processes, methods and IT tools will reflect and support issues found in existing sub-systems only. The new processes, methods and IT tools should therefore be part of the delivery and put in place before the project starts.
Unknown interfaces between the new components and the product lifecycle.	Knowledge about the interfaces of the new sub-system and the product lifecycle, e.g. testing equipment, manufacturing and assembly equipment, diagnostic equipment, servicing equipment, disassembly equipment needs to be built up [Ullman, 1997]. This can only be done by making a prototype vehicle with the solution fully packed and testing, installation, servicing and disassembly discussed with representatives from the respective domain.
Suppliers' lack of knowledge of OEM's processes and automotive standards.	The delivered processes and methods should either be harmonized with the suppliers' processes or the suppliers should harmonize their processes with those proposed by R&D or advanced engineering [Nobelius, 2002]. This means that the knowledge gaps regarding suppliers' processes and the suppliers' knowledge gaps regarding OEM's processes and industry standards are closed.

Long redesign loops in the verification process.	The relation with the suppliers of critical components should be managed less formally with increased focus on common visions and goals for the technology. Co-location with the supplier representatives in the development team is preferable [Johnsen, 2009].

It is important that the status of each recommendation is either resolved and/or agreed upon among the project leader and members, line department manager and members that own the new sub-system and the R&D or advanced engineering members. A clear status on each of the recommendations will reveal each knowledge gap and clarify the risks and thus also resources and time needed to be allocated in the project plan. Each explicit closing of a knowledge gap, appropriately documented, will also reduce the risk in the product development or industrialization process.

6. Future work

A method and tool for capturing and reusing knowledge about interfaces has been developed and is currently under evaluation by designers in the studied company.

Acknowledgements

The authors would like to gratefully acknowledge the Swedish governmental agency for innovation systems, VINNOVA, for financially supporting this work.

References

Adamsson, N., "Interdisciplinary Integration in Complex Product Development", Doctoral Thesis, Royal Institute of Technology, Stockholm, Sweden, 2007

Almefelt, L., Andersson, F., Nilsson P., Malmqvist, J.,"Requirement Management in Practice: Findings From an Empirical Study in the Automotive Industry", Research in Engineering Design, Vol.17., No3., 2006, pp 113-134

Bathelt, J., Jönsson, A., Bacs, C., Dierssen, S., Meier, M., "Applying the New VDI Design Guideline 2206 On Mechatronic Systems Controlled By a PLC", Proceedings of 15th International Conference on Engineering Design ICED '05, 2005, Melbourne, Australia

Bergsjö D., "Product Lifecycle Management – Architectural an Organisational Perspectives", Doctoral Thesis, Chalmers University of Technology, Göteborg, Sweden, 2009

Hull E., Jackson K., Dick, J., "Requirements Engineering", Second Edition, Springer, London, UK, 2005

Jansen, S., Welp, E. G., "A Heterogeneous Modelling Approach for Domain Allocation in Mechatronics", Proceedings of 16th International Conference on Engineering Design ICED '07, 2007, Paris, France

Johnsen, T. E., "Supplier Involvement in New Product Development and Innovation: Taking Stock and Looking To the Future", Journal of Purchasing and Supply Management, Vol. 15, 2009, pp 187-197

Kennedy, M., Harmon, K., Minnock, E., "Ready, Set, Dominate", The Oaklea Press, Richmond, VA, USA, 2008

Nobelius D., "Managing R&D Processes: Focusing on Technology Development, Product Development and Their Interplay", Doctoral Thesis, Chalmers University of Technology, Göteborg, Sweden, 2002

Ragatz, G. L., Handfield, R. B., Scanell, T. V., "Success Factors for Integrating Suppliers in New Product Development", Journal of Product and Innovation Management, Vol. 14, 1997, pp 190-202

Rahman, R. B. A., Pulm, U., Stetter R., "Systematic Mechatronic Design of a Piezo-Electric Brake", Proceedings of 16th International Conference on Engineering Design ICED '07, 2007, Paris, France

Ullman, D. G., "The Mechanical Design Process", 2nd Edition, McGraw-Hill, New York, USA

Ulrich K. T., Eppinger, S. D., "Product Design and Development", 4th Edition, McGraw-Hill Irwin, 2008, New York, USA

VDI 2006, "Design Methodology for Mechatronical Systems", Beuth, 2004, Berlin, Germany

Ziemniak, P., Marek, S., Stetter, R., "Mechatronics Engineering On the Example of an Innovative Production Vehicle", Proceedings of 17th International Conference on Engineering Design ICED '09, 2009, Palo Alto, CA, USA

Amer Ćatić, M.Sc., Lic.Eng.
PhD Candidate
Chalmers University of Technology, Product and Production Development
Hörsalsvägen 7A, SE-412 96 Göteborg, Sweden

Tel: +46 31 772 58 50
Fax: +46 31 772 13 75
E-mail: amer.catic@chalmers.se
URL: www.chalmers.se/ppd

Paper E

Knowledge management in mechatronic product development: effective method for creating engineering checklists

Ćatić, A. and Malmqvist, J. (2011)

Submitted for journal publication

Knowledge management in mechatronic product development: effective method for creating engineering checklists

Amer Ćatić and Johan Malmqvist,

Product and Production Development, Chalmers University of Technology

Abstract

This paper presents the results of an empirical study of a project whose aim was to develop a driveline for commercial vehicle applications containing a totally new mechatronic sub-system with a significant amount of new functions and interfaces. The purpose of the empirical study was to investigate which kinds of engineering knowledge that are critical in the development process of a mechatronic systems integrator. The results indicate that the most critical kinds of knowledge are related to component/system interfaces, failure modes, product functions and requirements. It was concluded that the knowledge regarding interfaces is poorly managed in the studied company.

Based on these findings it was decided to apply the method of engineering checklists to manage interface related knowledge. Guidelines on how to implement engineering checklists however are largely lacking and the main contribution of this paper is a method dedicated to creating engineering checklists from a knowledge management perspective. The evaluation of the developed method indicates a high level of both usefulness and usability of the method in the processes of the studied company along with potential improvement mainly regarding the instructions for the application of the method.

1 Introduction

The automotive business has been experiencing a trend towards specialisation both among the original equipment manufacturers (OEMs) and their suppliers. Simultaneously, a "mechatronisation" of the overall product has gradually been taking place with the aim to control the different sub-systems of the vehicle in a better and more effective way. The mechatronisation of the vehicle driveline has been guided by the aim to reduce fuel consumption and emissions while maintaining or even increasing the power output. These two trends have had a significant impact on the product development process of the OEMs where the detail design phase has been outsourced to the suppliers and the OEMs focus more on specification of requirements, integration of the supplied components and testing of complete system. The increased focus on integration and interaction between components has made component interfaces an important aspect of the product to manage in the processes of the OEMs.

The purpose of this paper is to propose and evaluate engineering checklists as a method for a more effective management of the knowledge in the processes of an integrator of a sourced mechatronic system. The paper is based on empirical findings from a studied product development project whose purpose was to develop a driveline for commercial vehicle applications containing a totally new and sourced mechatronic sub-system with a significant amount of new functions and interfaces (Ćatić and Malmqvist, 2010). The research question that has guided this effort is:

> *RQ1: Which types of knowledge are present in the development processes of a mechatronic systems integrator?*

Methods utilised both in academia and industry for managing knowledge in product development include lessons learned (NASA, 2010), knowledge based engineering (Stokes, 2001), design rationale capture (Wallace et al., 2005), best practices (embodied in e.g. design guidelines (Huang, 1996), handbooks (Pahl and Beitz, 1996) and standards (VDI 2221, 1993)). Within the framework of Lean Product Development the method of engineering checklists is successfully applied at the automotive OEM Toyota (Sobek et al., 1999). Little support however can be found in the literature regarding how to implement engineering checklists which, given their simple nature, might also seem simple to implement. In order to address this research gap and explore this method in the context of the studied OEM the following research question is posed:

> *RQ2: How can designers be supported in the creation and implementation of engineering checklists?*

2 Research approach

This research effort has been carried out in close co-operation with an industrial partner and has accommodated both academic and industrial perspectives and purposes. From the academic perspective, the research study was initiated with the purpose to investigate which types of knowledge that are present in an industrial product development project and process in order to propose and evaluate suitable solutions to manage the identified knowledge types. From the industrial perspective, the newly introduced mechatronic sub-system added a substantial amount of new functions, interfaces and suppliers to the product development process and organisation and the purpose of this study is therefore to identify knowledge that needs to be captured for future reuse in projects that will deal with a similar driveline architecture. The reason for choosing this particular development project was because it was perceived as a rather turbulent project with many schedule slips surrounded by much frustration. The research process is depicted in Figure 1 and summarised below.

The study was initiated with two workshops. One with the line manager of the department that is responsible for the development of the new mechatronic sub-system and one with the chief project manager for the development project. They were consulted for issues that had appeared

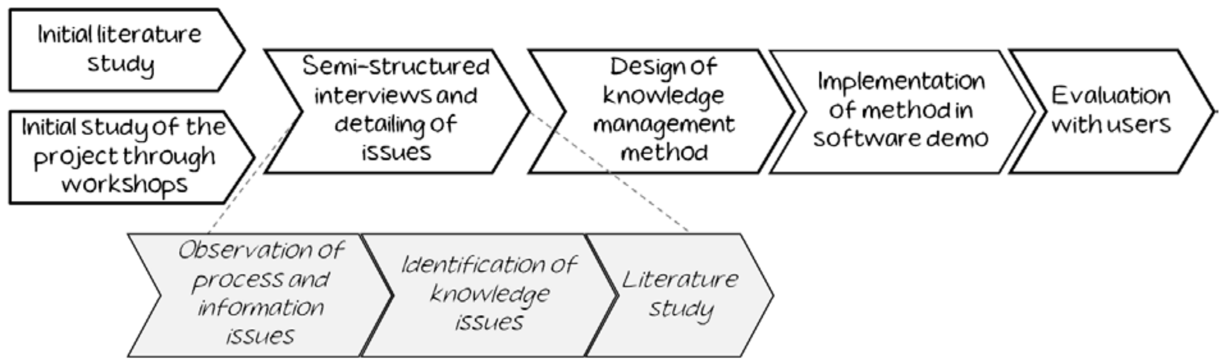


Figure 1: Research process

during the project and which they wanted to investigate closer. The focused issues were requirements management, component and system interfaces, the present "component-oriented" line organisation, and supplier management.

A total of 15 interviewees were chosen both from the line department and the project team, see Figure 2. They were interviewed in a semi-structured fashion and asked to reflect around the focused issues. The purpose of scoping a set of focused issue areas was to have the interviewees reason around these in great detail regarding the particular project as opposed to having them reason in general. As indicated in Figure 2 they were asked to focus on issues related to information management. The following information issues were observed:

- Poor quality of information (e.g. few details regarding alternative product solutions and decision rationale from the technology development phase and concept development phase of the new sub-system).
- Lacking information (e.g. no information on interfacing components, necessary components, supplier abilities or certain requirements).
- Wrong information (e.g. outdated requirements).
- Hard to find information (e.g. installation and assembly procedures or processes for parameter setting in supplied embedded software).

From the observed information issues a set of knowledge issues were identified as indicators of critical knowledge which needs to be managed. These results were the main outputs from the descriptive phase. All of the results found during the interviews were benchmarked against results in the respective fields found in the literature regarding mechatronics design, knowledge management, supplier management and requirements management. The detailed results from the descriptive phase up to this stage are reported in (Ćatić and Malmqvist, 2010). Results relevant for this paper are summarised in Section 4.

Based on the findings in the descriptive phase a method for managing the knowledge identified was proposed. After a review of knowledge management methods in the literature, along with those already applied in the studied company, the choice fell on engineering checklists. From the industrial perspective this choice was partly due to their alleged effectiveness and partly due to the fact that other methods such as lessons learned and design guidelines had already been tested

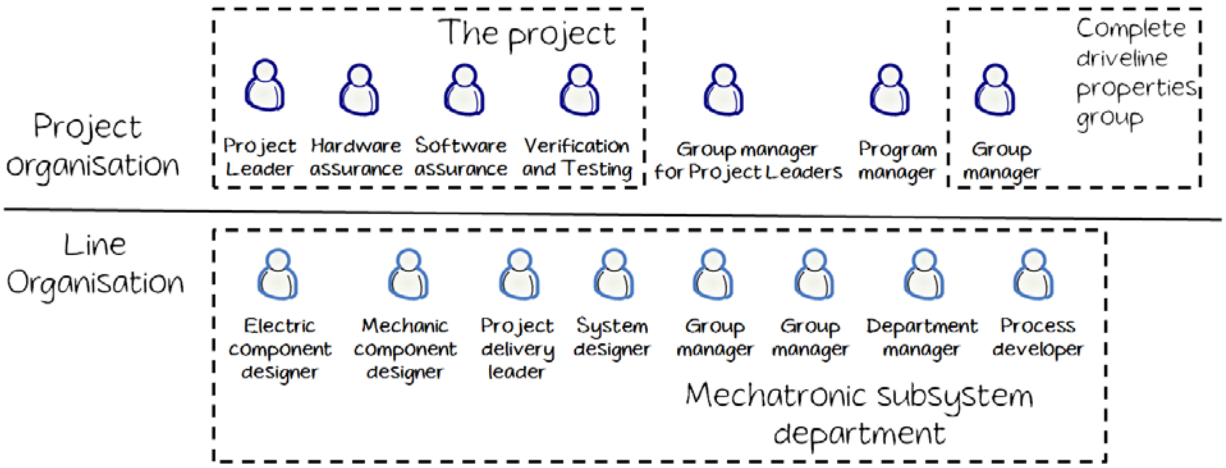


Figure 2: The interviewees placed according to their relations in the matrix organisation

with limited success in the studied company. From the academic perspective the reason for focusing on engineering checklists is the identified research gap regarding how the checklists can be created and implemented, see Section 3.2.

Given that the Research Questions 2 aims at investigating the application of a general method in a concrete case the method of a software demonstrator as a research method (Williamson, 2002) was preferred over interview or workshop based evaluation due to the need to investigate implementation related issues which are hard to foresee or imagine without hands-on usage. The method was implemented in a Microsoft Excel macro application and e-mailed to the individuals chosen for the evaluation. The macro application was accompanied by a document manual showing how the method and the macro application work along with examples to guide the users in using it.

The evaluation was performed in the form of short interviews (between 30-60 minutes) with the users. In order to triangulate the evaluation, the sample of test users consisted of individuals both from the initially studied department and from another department that is part of another division in the company. The test users were asked to reflect on the method's applicability in their engineering role and its usability as a carrier for knowledge. They were also asked to reflect on implementation related aspects such as how often the checklist should be updated, whether it should be implemented in some specific IT system and whether it was easy to understand and use.

3 Related work

The areas of mechatronic product development and engineering checklists are covered to give the reader an overview of the state of the art in order to delineate the identified research gap which are addressed through this paper. The third topic of knowledge categorisations is provided in order to explain the models applied for presenting the empirical findings in Section 4.

3.1 Mechatronic product development

In mechatronic product development a key issue is to ensure the most effective integration of the three involved domains of mechanics, electronics and software. Many research efforts are directed towards developing new methods and adapting existing methods to address the issues which arise in the integration of the three domains e.g. focusing on cross-domain interface and requirements management, roles and responsibilities, process and information management and verification and validation management.

Almefelt et al. (2006) have studied an industrial case and from this empirically derived a set of recommendations for requirements management which, among other things, address the need to early define and focus a set of over-arching cross-system requirements. In addition they also address the need to clarify each requirement with the underlying context and intent and define interfaces and verification methods for each requirement. Other contributions such as e.g. Jansen and Welp (2007) suggest models to primarily overcome the interface related issues by identifying and classifying different kinds of interfaces.

Adamsson (2007) addresses managerial implications of mechatronic product development and proposes increased awareness of the importance of the embedded software as well as increased need to organise for cross-domain collaboration. Another recommendation from Adamsson (2007) is to communicate that the product launch is not only about manufacturability but also validation of embedded software.

As a response to the need for specific methods and processes a VDI guideline called VDI 2206 (VDI 2206, 2004) has been developed which is based on the systems engineering methodology and addresses requirements and interface management issues along with verification and validation pointing towards useable IT tools for modelling and simulation. The guideline also gives a rough overview of how a mechatronic product is matured from an initial idea to production readiness. The research community has tested the method in several cases (Bathelt et al., 2005; Rahman et al., 2007; Ziemniak et al., 2009). A critique towards VDI 2206 is that it is recognised as the first systematic attempt to address mechatronic product development in an integrated way but essentially can be seen only as a framework of methods and tools within the three areas of mechanics, electronics and software (Vielhaber et al., 2010).

3.2 Engineering checklists

In the introduction, engineering checklists were mentioned as one of the less explored methods for knowledge management purposes. In the later years the paradigm of Lean Product Development, with Toyota as the main case in point (Morgan and Liker, 2006; Ward, 2007; Kennedy et al., 2008), has been on the rise. The use of engineering checklists is reported to be rather extensive in this paradigm as a way to carry knowledge in order to standardise processes and product solutions in areas with high certainty regarding the product or the process.

According to Morgan and Liker (2006) the engineering checklists at Toyota are highly visual and part or process specific (see Figure 3 for an example of a part specific checklist). A typical checklist can contain hundreds or more "checks". The key to managing this is constant revision of checklists in each development project, i.e. the checks are executed and revised simultaneously. Naturally this will add to the checklist which reflects the fact that the engineers are learning more and more about the product and the process. As constant revision and addition is part of the usage of the checklist a sense of ownership is achieved making the checklist not "another burdensome activity that has to be done" but something that constitutes a helpful and supporting tool in the process (Morgan and Liker, 2006).

Regarding the content of the checklists at Toyota, Sobek et al. (1999) state that each engineering function, such as e.g. styling, body, chassis and production engineering, defines "feasible regions" for their product or production solutions based on past experience, analysis, experimentation and testing. These feasible regions are translated into engineering checklists or design standards. Each checklist is divided into a number of areas such as functionality, manufacturability, government regulation, reliability and so on. Sobek et al. (1999) give the following examples for how a "check" can be formulated:

- *Piston rings of standard material should have thickness of at least 1.8 mm to provide proper seal.* (Functionality)
- *Bounds on acceptable curvature radii for sheet metal bending.* (Manufacturability)
- *Minimum strength characteristics for door members to meet side impact crash tests.* (Regulation) (Sobek et al., 1999)

The checklists do not only contain design guidelines but also information on what can and cannot be done in an economical fashion or how to incorporate new technologies for automation, cost reduction, quality improvement and so on. The authors state that in the very early stages of a vehicle program engineering functions exchange their checklists with each other and with the chief engineer of the vehicle program to update each other on what is possible since the last vehicle program in terms of new technologies and solutions. Such a knowledge transfer between any two engineering functions whose sub-systems interface or interact is very important in order to prevent assumptions regarding what one sub-system can expect from another sub-system.

Engineering checklists are used within other areas of engineering as well. In the field of production systems development checklists are used to ascertain that relevant parameters of safety, health and ergonomics have been considered in e.g. work station design (Munck-Ulfsfält et al., 2003). A similar approach can be found in the eco-design area for environmental considerations in product development (Tischner et al., 2000). Checklists can be found in the field of new product development as knowledge carriers from academic publications to industrial practitioners. They are however rather general and cover topics at a high level of abstraction such as strategic planning, market research, idea generation, idea screening and evaluation, product planning design and engineering, manufacturing planning and so on (Ribbens, 2000). The contributions which have elaborated the furthest around the creation and use of checklists

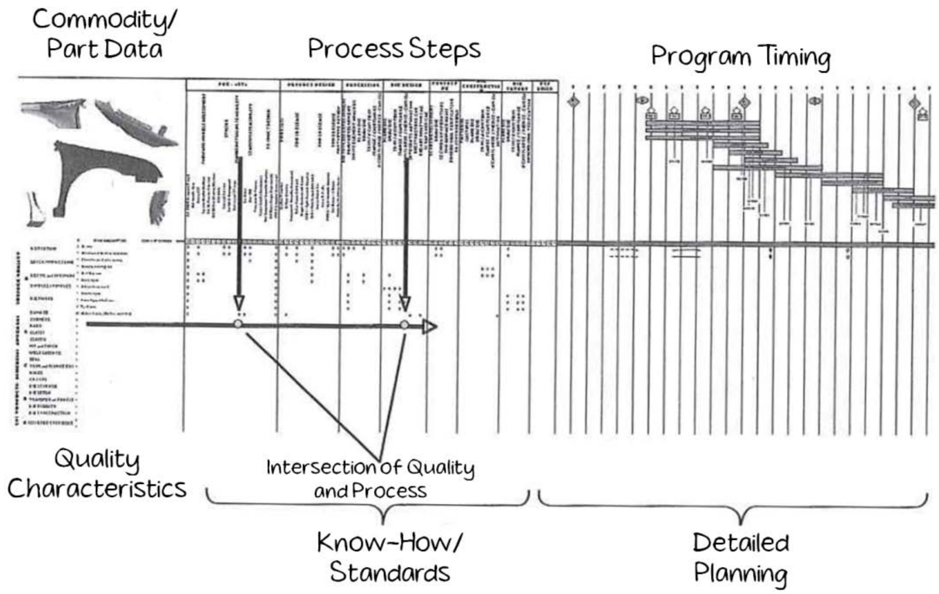


Figure 3 – Example of engineering checklist for a part from Toyota (Morgan and Liker, 2006)

applied in software development for the purpose of inspection to verify the correctness of software documentation and software code are provided by Kokkoniemi (2002; 2006). More specifically Kokkoniemi (2006) has investigated the implementation process of how to introduce the practice of engineering checklists into the software engineering process for the purpose of software inspection. Kokkoniemi describes three different approaches for generating checklists:

1. **Literature-adopting approach**
   Checklists are copied from the literature and slightly modified before use
2. **Consultant-based approach**
   Checklists are generated by a person who knows the software development model and practices of the company
3. **Workshop-based approach**
   Checklists are generated in a workshop prior to usage by those who will use them.

From Kokkoniemi's (2006) experience the first two approaches did not work very well. The literature-based approach did not work because the software development model in the company focused on different details than those in the literature. The consultant-based approach did not work either because the checklists need to be based on the experience of those who are performing the software inspection in order to be relevant. Finally the workshop based approach was found to generate valid checklists because they are based on experiences from the inspection process. The role of the workshop itself was that it constituted the method for knowledge extraction. Based on his experiences Kokkoniemi has compiled a set of recommendations for implementing checklists (recommendations that reflect software inspection specific issues have been omitted in the following list):

1. Checklist must not be too long
2. Checklists that try to consider everything are impossible to make to work
3. If checklists risk of to become too long, they should be split into several lists
4. Checklists must be tailored separately for every organisation
5. When checklists have been generated they must be inspected

These recommendations are somewhat in conflict with the findings presented by Morgan and Liker (2006) especially regarding the aspect of how many items that should be present in the checklists. Kokkoniemi (2006) is of the clear opinion that they should be kept short while Morgan and Liker (2006) state that at Toyota the checks in a checklist are counted in hundreds. From the perspective of this paper however the fourth and fifth recommendation by Kokkoniemi are of higher importance. The fourth recommendation implies that in order for checklists to be useful they have to reflect the unique characteristics of the organisation while the fifth recommendation gives support to the need for some kind of “release process” for the formalised knowledge that is managed in the checklists.

3.3 Knowledge categorisation

In order to orient the reader in the model applied for the purpose of categorising the identified types of knowledge in the empirical findings this categorisation is described in more detail in the sub-section. The model combines the classic know-what, know-why, know-how and know-who framework from (Lundvall and Johnson, 1994) together with Sunnersjö’s division of knowledge into product knowledge and process knowledge (Sunnersjö, 1994). The categorisation model thus obtained is illustrated in Figure 4.

3.4 Research gap

The literature in the field of development of mechatronic products covers many different areas and aspects of mechatronic product development. The contributions described have been focusing on managing embedded software in different ways (Adamsson, 2007; Bergsjö, 2009), applying a dedicated methodical framework for developing mechatronic products (Bathelt et al., 2005; Rahman et al., 2007; Ziemniak et al., 2009) or dealing more or less purely with requirements management (Weber and Weisbrod, 2003; Almefelt et al., 2006). The aspect of knowledge in the mechatronic product development process has been addressed only through an

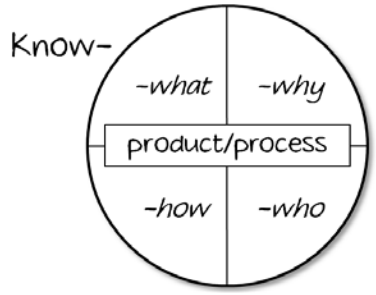


Figure 4 – Knowledge categorisation model

attempt to build an IT tool for computer aided mechatronic design with inclusion of explicit knowledge for the purpose of automation (Counsell et al., 1999). Our earlier contributions focus on what kinds of knowledge gaps to expect and how to manage them when initiating a new product development project with a new mechatronic sub-system (Ćatić and Malmqvist, 2010). Based on this we conclude that there is a research gap regarding empirical studies describing mechatronic product development processes from a knowledge perspective. In addition, there is also a lack of prescriptive contributions regarding methodological support for knowledge management in the context of mechatronic product development.

In respect to the topic of engineering checklists as a method for knowledge management the literature from the field of product development suggests that the use of engineering checklists for the purpose of knowledge management in product development is beneficial and refers mainly to Toyota as the main case in point. Simultaneously this literature provides little support in prescriptive statements on how to go about creating, implementing and maintaining checklists. Some support in this concern can be found from the field of software engineering but the lack of product development aspects are lacking since the contributions mainly deal with software inspection and not software development. Based on this we conclude that there is also a research gap regarding methodological support on how to create, implement and maintain engineering checklists in a product development context.

4 Empirical findings

This section covers only the findings which are related to knowledge management in the studied case and that have been used as the main background for the design of the method described in next section. For a more detailed and complete summary of all findings related to the studied case see (Ćatić and Malmqvist, 2010).

Development projects in the studied company are organised according to Figure 5. There is a product planning organisation which officially places an order to a driveline development project which consists of 15-20 people who plan the project and in their turn place an order to the line departments which are structured according to the sub-systems of the driveline. The studied driveline has a different architecture compared to previous ones with a new mechatronic sub-system. The addition of the new mechatronic sub-system has also meant that a completely new department was established in the organisation. This department had no explicitly defined processes and the studied development project is their first. The consequence is that they have had to built up a large amount of new knowledge regarding both the product and the processes during the studied development project and this knowledge needs to be taken care of in a structured and strategic manner.

As indicated in Figure 5 and also noted by others in the automotive field (Weber and Weisbrod, 2003; Almefelt, 2005) the OEMs role in the product lifecycle has become to act as system integrators of supplied systems. This is especially true for the studied new department. The focus

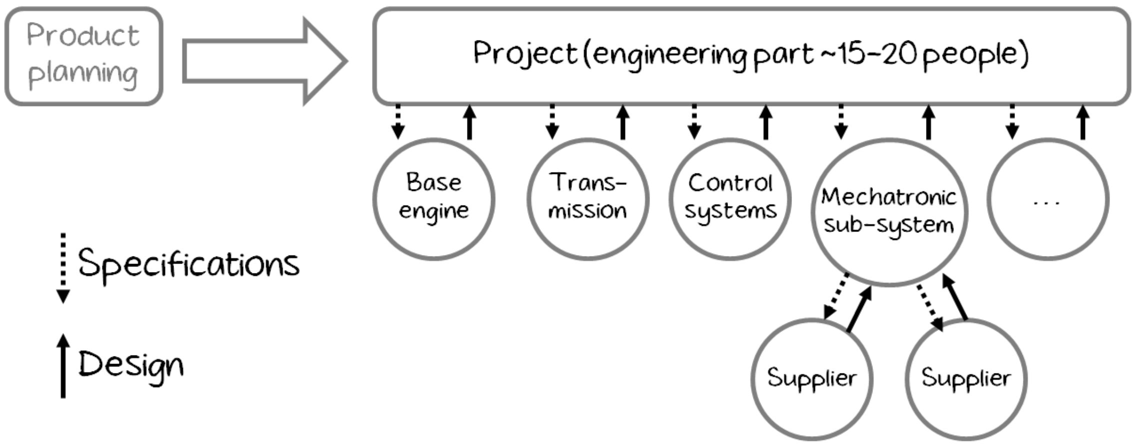


Figure 5 – Organisation of development projects at the studied company

of the engineers working in this department is to receive the requirements coming from product planning and the project organisation, interpret those into requirements posed onto each of the components in the mechatronic sub-system along with verification methods for those requirements documented in a specification that is sent to the component supplier. Each of the components has a component owner who does this work. A note of clarification: technically many of the components are mechatronic systems of their own since they are mechanical but also have embedded control circuits and software over which the component owner at the OEM has little control. These circumstances make requirements management and interface management critical for success, as was concluded in our earlier contribution on this case (Ćatić and Malmqvist, 2010).

4.1 Product knowledge

In the category of product knowledge the following types are identified to be important in the studied case:

1. **Interfaces**
    a. Know-what interfaces a component has in the three domains as well as in the lifecycle
    b. Know-who is responsible for each interface
2. **Requirements**
    a. Know-what requirements there are
    b. Know-who is responsible for which requirement
    c. Know-why the requirements are as they are (i.e. the rationale of each requirement)
    d. Know-how the requirements are verified
3. **Functions**
    a. Know-what functions a component contributes to
    b. Know-how the functions are verified
    c. Know-why certain functions exist
    d. Know-who is responsible for each function

4. **Failure modes**
    a. Know-what failure modes a component contributes to
    b. Know-why a component contributes to certain failure modes
    c. Know-who is responsible to make sure a failure mode is addressed. Is it the person in testing who detected the failure? The component owner whose component failed? Or the component owner whose component was the root cause of failure?
    d. Know-how failure modes are detected, e.g. some sub-systems experience failure modes at revs per minute (RPMs) half of maximum and not at maximum

There exists a tool in the company (implemented in a commercial content management system) for managing design guidelines. The aim of that tool is to support the management of the above stated knowledge but it is clear from the respondents that it does it poorly and the tool is not used at all. The structure of the tool is such that each component owner is supposed to write a guideline containing the requirements with belonging methods and tools for verification, functions and failure modes of the component with links to error reports. The knowledge about interfaces is not addressed in any way even though much of the requirements and failure modes, and even some functions, all are directly related to various types of interfaces. The tool is also not supported by any methods on how to write a design guideline. The only guidance is given by a list of topics which need to be covered and a few examples of design guidelines for some (mostly mechanical) components.

Some of the interviewees from the project organisation were engaged in this project because they are considered as experienced designers who previously worked as component owners in other sub-systems. It was clear that these individuals kept all the different stated types of knowledge implicit in their head and it was part of their working procedure as opposed to the new department where the component owners were not only new to their components as such but also to the organisation. This lack of experience rendered them less efficient even though each and every one of them was rather experienced in their respective field of expertise (e.g. high voltage DC to low voltage DC converter specialist).

4.2 Process knowledge

In the category of process knowledge the knowledge about which information (and from which sources) is needed in different forms is mostly prevalent. Some critical knowledge related to tools, such as special simulation tools, was also present for some component owners. Knowledge about activities and their sequencing was however troublesome to define. The reason for this could be that it is deeply embedded in the individuals and therefore hard to externalise. Or it was simply non-existent due to a reaction based, and therefore unstructured, approach in the project. An overall personalisation strategy for managing process knowledge is implicitly adopted in the organisation which means that the critical process knowledge is “know-who”. This was confirmed by the fact that knowledge about who has which knowledge could easily be externalised as an answer to a direct question to the interviewees.

4.3 Identified need for knowledge management in the studied case

The primary need in the studied process and organisation is to capture knowledge related to interfaces. The lack of knowledge about interfaces and how these are managed along with which information is needed in relation to each interface was believed by the component owners to have had the largest contribution to delays and schedule slips in the project. This is a consequence of the fact that it is a completely new sub-system with many unknown new interfaces and a completely new department with new people and no established processes. From the company's point of view it is of critical importance to capture and reuse as much of this knowledge as possible to avoid a similar situation in future projects and to speed up the establishment of processes and practices in developing the new sub-system. With this as background and in relation to the already existing system for design guidelines it was concluded that from the perspective of the particular company there is a need for a method which will support capture and management of knowledge related to interfaces.

5 Method for creating engineering checklists

In order to meet the needs to capture knowledge in the studied case the following method is proposed and evaluated. The aim of the method is to ease the creation of engineering checklists as knowledge carriers. The method is based on the visual appearance of Ishikawa diagrams and its purpose is to act as a stimulator to capture the knowledge that will go into the checklist.

5.1 Stage 1 – Pre-study

The aim of the pre-study is to clarify the level at which engineering checklists are to be created and how often they should be used and updated. From a knowledge management perspective this translates into positioning the checklists at a level where the context of knowledge creation is as similar as knowledge reuse. From an implementation perspective the purpose of this stage is to align the checklists with the way processes are executed so that the engineering checklists are perceived as purposeful by the users.

In the studied case the appropriate level for the checklists was found to be at the component owner level. This is due to two aspects. Firstly the component owner role reflects the way work is divided and organised. Secondly the component owner level reflects a rather stable form of organisation. One should not confuse the name component with a single part in the product assembly, many of the "components" are in fact small mechatronic systems with own embedded software and control circuits. If positioning the engineering checklist at a lower level (e.g. one component owner has three checklists for each of the domains of the current component) they would risk to be far too contextual and specific for each project, affecting the reusability of the knowledge in a negative way. On the other hand if positioning the checklists at a higher level of e.g. sub-systems would result in too little contextualisation and risk of the checks and guidelines to be too general, for example "when designing the control system it is important to have a preliminary view of the torque and power curve of the engine". It is hard to provide a general guideline at which level the checklists should be positioned. It should be iterated with the organisation considering the mentioned aspects to ascertain a proper level of implementation. In addition to the level at which the engineering checklists are positioned it is also important to consider how the checklists are integrated with the process. When and how the checklists are used and updated depends on how the engineering processes are set up.

5.2 Stage 2 – Introduction of engineering checklists into the process

This stage is concerned with the initiation and making of the first checklist. The important thing to focus on in this stage is not the completeness of the first checklist but a proper understanding, on the part of the designer, for how the method is executed and how it is supposed to support the process. The reason why completeness is traded-off in the first iteration is because the checklist will be updated with new checks every time it is executed. Therefore, with time, a "completeness" will be achieved. The first argument for adopting such an approach is that a pursuit for completeness is time consuming and negatively affects the impression that the method is quick and easy to use. The second reason is that it is unreasonable to expect a complete set of valid checks at once due to a limited capacity and focus of the human brain. The following three steps are carried out in the introduction of the method.

Step 1 – Mapping out interfaces, inputs and outputs

In the first step the designer is asked to think about the component, sub-system or function he or she is responsible for and provide all the known interfaces (geometrical, functional, physical, informational) and map out corresponding information inputs needed and outputs produced anywhere in the development process, see Figure 6. It is important that the time aspect or sequencing is not considered. Our personal experience is that designers have a difficulty defining sequences between their activities since there are many unplanned loopbacks and if-then scenarios with unpredictable situations. A similar approach is advocated in the confidence mapping between parameters in the "Signposting" method (Clarkson and Hamilton, 2000) for process modelling. The only sequencing that should be considered is between inputs and outputs. The difference between inputs and outputs is that inputs are related to interfaces that information is needed from and outputs are the interfaces that information is provided to.

Step 2 – Mapping out causes of lead time

For each of the inputs and outputs the designer should reflect on issues regarding that input which contribute to lead times. It could be uncertainty, limited availability of responsible people, process issues (e.g. that the input is available very late in the process), ambiguity due to redundant information sources and so on. The interviewed designers had a weaker perception of such issues related to outputs but there were some exceptions where they knew that e.g. "this piece of information, even if only preliminary, should be sent to the simulation group as soon as it is acquired because it is critical for the configuration of their simulation models".

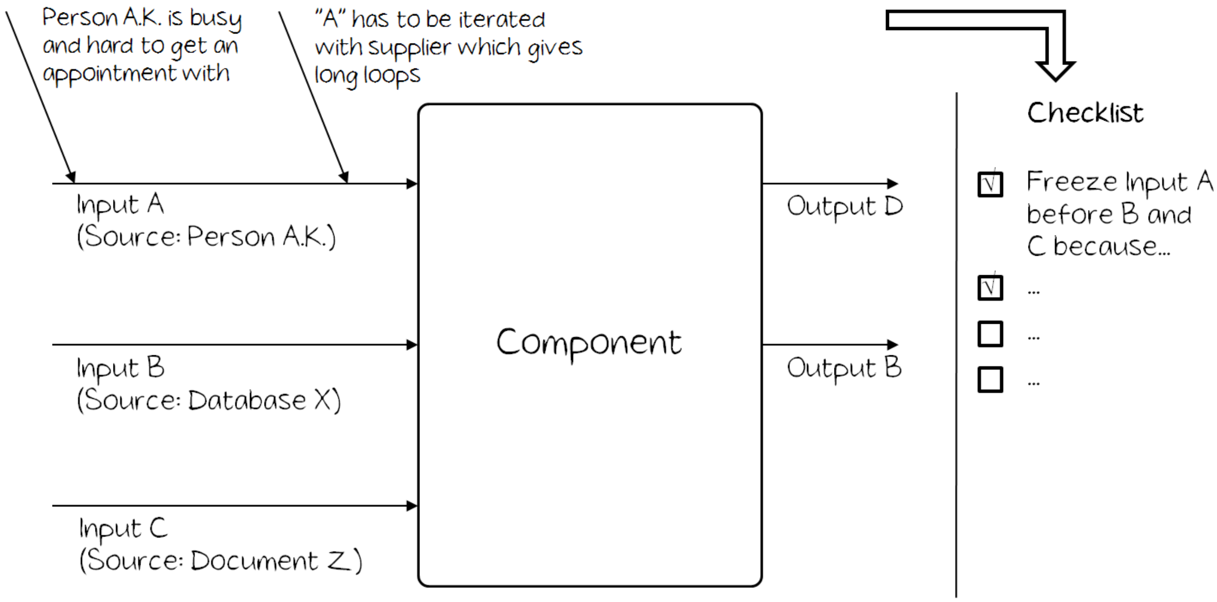


Figure 6 – Steps of the method

Step 3 – Making the first engineering checklist

Once the inputs, outputs and factors affecting the lead time related to them are mapped, it is time to derive the checks for the first version of the engineering checklist. A guideline is written as a check for each input regarding the input itself and regarding causes of lead time and likewise for the outputs. Even though the first step, the pre-study, should largely cater for a correct contextualisation it is important that each designer writes the guidelines in the

checklist keeping in mind that it should be both concrete enough and general enough to be reusable. To a certain level, the design of the method addresses this issue by making the designer reflect around a component or a role which provides a context for the guidelines but generalised across projects. In the evaluation of the method, the designers were provided with an instruction that guided them through the method with examples, hints and recommendations on how to think.

5.3 Stage 3 – Use of the engineering checklists

Before the first engineering checklist (established in the previous stage) is used for the first time, the issue of process integration, mentioned in Stage 1, needs to be considered. If the development process and projects are structured according to a stage gate model one alternative is to connect certain checks to each gate and make the checks as reminders of important things to consider or activities to perform in relation to each gate passage. In the studied company a general purpose checklist related to product documentation and integrated with the stage gate process exists. The checklist works as a general reminder to e.g. update requirements lists, check-in the latest drawings to the corporate PDM system, update document versions, update the project status reports, initiate purchase processes, sign contracts with suppliers and so on. Another alternative is to keep the checklist on the designers desktop as an everyday reminder of the total amount of activities to do or aspects to consider without implying an order of sequence but stating that certain total of checks should be addressed before a certain gate. The important aspect to consider, when thinking about how to integrate the checklists with the process, is that their execution is as aligned with the process as possible to provide proper support. If this means that the checks are executed and updated at every project gate, at each component release or according to a time schedule (e.g. every three months) needs to be resolved on a case by case basis.

Regardless of exactly how the checklists are integrated with the processes it is important that they are executed and simultaneously updated in some defined intervals. The reason for this, besides process integration, is to also better support the knowledge capture process. By periodically updating the checklist, solutions to recent issues, fresh in the mind of the designer, are captured and included in the checklist. As opposed to updating the checklist at the end of each project when many of the early issues and solutions are forgotten the periodical updating of the checklist caters for a continuous knowledge capture. Therefore, in addition to process alignment, it could be beneficial to reason about probable knowledge creation cycles in the development process and consider those as well.

5.4 Usage of results and motivation

The idea is that the usage of the checklist and the diagram, depicted in Figure 6, should primarily be beneficial for the designers themselves to ensure that important issues are considered. Morgan and Liker (2006) note that one large benefit of checklists is in the introduction of newcomers who learn about the design work. In addition the authors also note that designers at Toyota were not only knowledgeable of their own checklists but also of other designers' checklists, such as those of interfacing components or manufacturing engineers concerned with the manufacturing operations related to the component in question. This made them aware of which information they produced that was critical to others and could facilitate better cooperation and communication.

In addition to benefitting the designers other roles that could benefit from the results of the proposed method are process and IT owners whose responsibility is to ensure that their processes and IT systems provide proper support to the business processes. These can analyse the diagrams of a complete department or group of designers to identify critical information flows and use them as input for process and IT improvements.

There is a risk that the designers do not see any large benefits of using the proposed method. To offset this risk the method was made quick and simple in order to reduce the time needed for its execution. In order to further increase the motivation to continually revise the checklists it would be beneficial to use the diagrams in the discussions regarding process and IT improvements. This would make it apparent that the documentation from the method is used and leads to concrete actions in business process improvement.

5.5 Implementation of the method

In order to increase the scalability of the method and enable an easy distribution to many designers the method was implemented as a macro application in Microsoft Excel. It is however in no way limited to this form of implementation. It would be more beneficial to use e.g. a web-based solution implemented as a an internal web-page that is logged on to and connected to a database for management of checks and checklists as individual objects rather than files. Such a solution would enable to gather and connect all the checklists from a department for a more holistic analysis of critical information chains and interactions. A business process improvement initiative could then apply the notion of "information pull" to rearrange the processes and organisational setups in order to optimise these flows of information and ease interaction. This is harder to do in a file-based setup as the one implemented.

In addition to the macro application a short guide was appended to instruct the designers in executing the method. As already discussed, the importance of this guide is critical because it ensures that the application of the method is as universal as possible. Otherwise there is a risk that e.g. inputs, outputs, lead time causes or resulting checks are at varying levels of abstraction making it hard to compare and connect different checklists and models. One designer may write that he/she uses "the project specification" as input while another may write exactly which requirements from the project specification that constitute inputs for his/her component. In the validation of the method the user's guide was written in a Microsoft PowerPoint presentation consisting of three slides (one slide per step) explaining what should be done in the three steps, how the designer should think with lots of examples at an appropriate level of abstraction. Throughout the guide the designer is asked to not think about sequencing and processes but think from the perspective of his/her component/sub-system and think about which interface related information that is needed for the design of the component/sub-system.

Besides an easy to use IT application of the method and a short and comprehensive guide with lots of examples the third aspect of the method implementation is time spent on execution. In each step of the method guide it is stressed that there is no need to spend a lot of time in trying to map out the complete set of inputs, outputs or variations in the first execution of the method. The mindset should be to do this from the top of the head, and a recommended total time for instructions + method was 30-40 minutes. The reasons for this are two: it should not be burdensome to execute the method and, since the checklist is updated periodically during the design process, anything that is left out in the first iteration will surface in later iterations.

6 User evaluation of the method

The two files containing the macro application and the method guide were sent out to 10 engineers. In order to evaluate the method for all kinds of engineers five of the engineers were designers belonging to the originally studied department and five engineers working in the same company but at another division. The main difference between these two groups was that those originating in the studied department are organised as "component owners" each with a clear responsibility for a certain component in the mechatronic sub-system. The control group on the other hand was responsible for an onboard system in the vehicle which mainly consists of software. This group consists of roles responsible for certain characteristics and functions, testing, documentation, versioning and so on. The reason for the two groups is to evaluate the method in another type of organisation than the one from which the needs for the method were derived.

The respondents were urged to spend preferably between 30-40 minutes and not more than 60 minutes on the execution of the method. They were supposed to execute the method on their own using only the appended guide as support. A subsequent interview with the respondents was set up where they were asked to reflect on the usefulness (in terms of how useful the results of the method execution are), the usability (in terms of ease of use of the method itself) and suggestions for improvement. In addition, the respondents were asked questions regarding how much time they spent on the execution, the frequency with which the checklist should be updated and who else, apart from themselves, would be interested in accessing the results from their diagrams and checklists.

Response rate

In the department, from which the original requirements were derived, only two out of five engineers responded. Out of the three that did not respond, one was a consultant that moved to another company and two were told to prioritise highly critical tasks due to budget cut-backs caused by the global financial crisis. In the other department the response rate was 100% with all five respondents taking the time to execute the method and participate in a follow-up interview, four of them were software designers and one was a software development project manager. The total of respondents is therefore 7 engineers: 1 mechanical designer, 1 mechatronic designer, 4 software designers and 1 project manager.

Reflections regarding the use of the method

All seven of the respondents found the usability of the method to be high. Most of the comments were directed towards the simplicity and ease of use of the method which were credited to the visual appearance and graphical approach, see Figure 7. The five engineers dealing with software also commented that they now had collected their inputs and outputs in one place which is something that today does not exist. Most of the process maps or guidelines are at a too high level of abstraction to be useful as carriers of knowledge (mainly process knowledge) for specific day-to-day activities. Two of them responded:

> "*It is a good thing, I complained a lot about the lack of defined inputs when I started here 1.5 years ago.*"

> "*The method works very well since it focuses only on inputs and outputs – the external factors. Without clear guidance it is easy to start assuming things both in what is needed but also about constraints and boundaries.*"

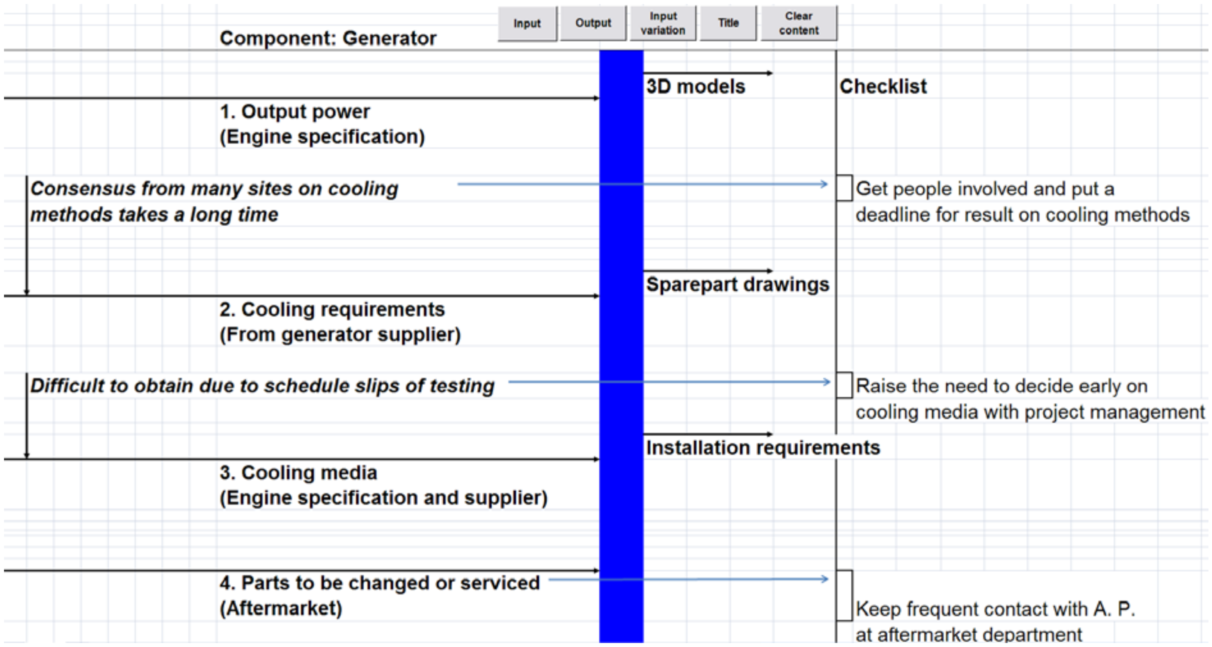


Figure 7 – Screenshot of the macro application demonstrator

A side effect of executing the method that was stated by three of the respondents was that they generally found it rewarding to simply do a reflection of how their work is carried out. They stated that this benefitted them personally. One of them commented:

> *"It gave me a good overview of my situation and some very concrete points came out as 'advice for myself'. I was surprised given the simplicity of the method."*

Finally all seven would recommend their colleagues to use the method as part of their process improvement work.

Reflections regarding the usefulness

All seven of the respondents stated that the results coming out from the method (the diagram and the checklist) provide useful information and support for the processes. Most of the respondents stated that the results were useful for several people. Four stated that their future replacers or newcomers had the most benefit of the results. Five stated that they themselves could use the results but for different purposes. Three of them commented that keeping a list of things to remember was purposeful for them. Two said that the results could replace the existing "lessons learned"-documents and that the method can be used to both support a structured approach for writing lessons learned and simultaneously support the documentation of lessons learned. One of them said:

> *"I think this is a very good and structured method for writing lessons learned after project closing or dedicated testing…Usually lessons learned are not a structured approach and you don't know if you caught all the lessons learned. This way you get a context for them and a better overview."*

In addition one thought that those who affect how work is done such as project leaders, process owners and IT owners could benefit from using the different diagrams/checklists as input for process and IT improvement.

Reflections regarding the integration of the method with the processes

The average time for the first execution of the method was 45 minutes including reading of the appended guide. The respondents had different ideas for when and how often it would be suitable for them to update the diagram and the checklist. The differences reflected each of their own unique and individual situations. All of them stated that the method should be executed in the end of each project. One said that, in addition to this, an execution once a month was appropriate while one suggested 2-4 times per year. One stated that the diagram should be updated after each project (reactively) and a checklist should be created before each new project to proactively address the issues in the specifics of the new project.

Reflections regarding shortcomings of the method

Most of the negative comments were related to the instructions for the method. The instructions did not clearly distinct between the fact that the method could be executed in a general way and in a project specific way. One of the respondents commented:

> "*Unclear which the intended role is. Project leader in general or project leader for a specific project.*"

Executing the method for a role "in general" provides a more general diagram and a checklist which is closer to the process maps that already exist in the organisation. Another respondent commented that the method does not reflect reality, stating that the model is "over-simplified" while the other respondents appreciated the fact that the method is simple, focusing only on inputs and outputs. The first respondent was asked to elaborate the comment to pin-point the exact issues which contributed to the sense of over-simplification. After some discussion it was clear that the main issue was a perceived implication of linearity in the work, i.e. the respondent perceived the method to assume that the conversion of inputs to outputs was done in a linear fashion. The main critique was that a lot of the outputs also constituted inputs in an iterative process and that the method should be able to reflect this.

Ideas for improvement

The respondents were also asked to provide some ideas for improvement of the method and the respondent who gave the remarks regarding the implied linearity proposed that the implementation of the method should allow for outputs to be looped around as inputs. Most of the other suggestions were related to the way the method is implemented. The stated ideas were:

- To create a clearer and closer connection between the checks in the checklist and the inputs and input variations.
- Possibility to divide inputs into categories to gather inputs based on which project/process phases they belong to.

- Possibility to model the mechanisms involved in the conversion of the inputs into outputs

Conclusions regarding the evaluation

The evaluation has showed that most of the expected characteristics with the method, such as ease of use combined with useful results, have been achieved. The expectation that the results would be useful and benefit both the designer, in terms of supporting continuous improvement, and others, such as newcomers and support functions (process and IT owners), was confirmed.

One of the major shortcomings of the method as it is currently implemented is the inability to support the modelling of iterative activities. In addition the instructions for the execution need to be further clarified in how the inputs/outputs are modelled in regard to roles and project-specific versus general inputs/outputs.

The overall conclusion is that the method is applicable and useful in the context of a system integrator. Its ability to capture relevant and useful knowledge in an easy and not so time consuming way can motivate its use, both from an individual perspective and a corporate perspective.

7 Discussion

In this section the contributions of this work are discussed from the perspectives of the two research questions posed. In the first subsection the empirical findings are compared with other findings regarding the characteristics of the mechatronic product development process. Furthermore the subsection discusses the implication of the component oriented organisation on knowledge management in the studied case. In the second subsection some reflections are provided regarding the design and evaluation of the method for creating checklists described in Section 5 and 6 mainly from the perspective of Research Question 2.

7.1 Discussion of the empirical findings

With respect to Research Question 1, whose answer can be found 4.1 and 4.2 it is clear that the empirical findings show large similarities with other findings that have been observed in the development of complex and mechatronic products which are mainly sourced (Almefelt et al., 2006; Bergsjö, 2007). A high focus on functions and requirements is required and puts an increased emphasis on how the components of the system interface but also how they interact and produce desired, as well as undesired, system effects. As is evident from the knowledge related findings these product characteristics are rather well reflected in the types of knowledge identified as important in the process.

The management of this knowledge, however, is more of an issue that can be attributed to the component oriented organisation which has some negative implications on the possibilities to both learn from errors and reuse the knowledge to avoid them. The component oriented organisation is rather effective from a supply chain management perspective since each supplier has only one contact person at the OEM who is responsible for specifying and verifying the component, managing the component documentation and assuring all necessary contracts are signed in the purchasing process. The effects of the component oriented organisation (that have a negative impact on the knowledge management) are two. The first effect is that system level design is separated from component level design meaning that the two levels (system level engineers and component level engineers) have to be able to communicate via fail-safe requirements specifications otherwise there is risk of both redundant requirements (with the risk of them differing) as well as lacking requirements (both of which were observed). The second effect is that system level failures are separated from component level failures meaning that a component can fulfil its specification but fail in the system due to system effects. The consequence of these effects is that there are barriers regarding the flow of information between these two levels, especially since the levels are also separated in the organisation. For example the information that one component failure is caused by some other component's behaviour might not reach the owner of that second component owner and the knowledge created from this failure might do more harm than good in the future.

With this in mind it should be explicitly stated that the method of engineering checklists may not be the most appropriate method for managing knowledge in this particular organisation. As mentioned in our previous contribution (Ćatić and Malmqvist, 2010) in order to address the knowledge management issue properly the component oriented organisation needs to be complemented with e.g. a function oriented organisation to strike a better balance between the two aspects of integrated system/component level design and the supply chain management in the product development process. Since this was not an option (at least not in the short term) for the company in question the method of engineering checklists was found to be suitable as a way to overcome some of the worst information barriers and simultaneously reduce the time spent on interface meetings to enable a more effective knowledge management.

7.2 Discussion of the method for creating engineering checklists

Proper contextualisation is an inherent challenge of all knowledge management initiatives which are based on any level of codification of knowledge i.e. they tend to document specific solutions to specific problems (Fernie et al., 2003). The usefulness of such knowledge management methods degrades as similar symptoms reoccur in projects due to similar, but not identical, problems since those whose knowledge is codified seldom make a root cause analysis, to generalise both the problem and the solution. Later, the solutions seem as non-applicable to those who try to apply them as observed by e.g. Rinman and Wilson (2010) in their study of a lessons learned implementation. Dixon (2000) labels this as an inability to formulate knowledge based on experience. The first execution of the proposed method in Section 5 should contain both general issues and issues found in a specific project while subsequent executions should add to the existing base with new issues found in new projects as they are executed. This might sound very easy but there is a challenge which, if not addressed properly, might render the results from the method of limited value. The challenge lies in the formulation of the guidelines in the checklist at the right level of abstraction regarding the context of application.

The comments regarding “general” and “project specific” problems by the respondents could be an indication of this issue. This has been addressed by implementing the checklists at a level (component owner) that ensures the narrow context of a specific component/sub-system. As opposed to e.g. lessons learned which are usually written in the context of a complete project with many details that might differ across projects (Schindler and Eppler, 2003). Nevertheless, it is important to consider this risk and behaviour when writing the instructions for the method and educate the users to highlight this issue.

Another issue that is highlighted in the evaluation but is more related to the implementation than to the way the method works is how the method is integrated in the processes and which roles are supposed to use it. It was clear that the respondents had different views on how and when the method should be used, therefore it is important that this be clarified, preferably in use-cases before an implementation, because it seriously affects how the method embraced by the organisation.

The idea of checklists is that the processes in which they are applicable should be of a repeatable nature. This limitation is also valid for the proposed method since the concept of inputs and outputs assumes a process which should be at least converging towards a delimited set of possible activities. This limitation is not a big one but it needs to be considered. In other terms the method is applicable for processes whose main focus is not innovation but lead time reduction i.e. going from a set of fairly set requirements and a chosen concept towards an industrialised product as quickly and with as few errors and rework as possible. In this respect repeatability is sought for since the industrial structure with supply chains and production facilities is not likely to change very often or very much.

With respect to this the method is not delimited to being useful only for system integrators such as the company studied in this case. It is however apparent that system integrators’ main challenge is to document and manage interfaces to keep track of each of them in the integration phase in order to be able to test and optimise the system. In this context the usefulness of the proposed method is high.

8 Conclusions and future work

In respect to Research Question 1 it can be concluded that a mechatronic system integrator needs to manage product knowledge related to requirements, interfaces, functions and failure modes as well as process knowledge related to the information flows surrounding the product knowledge. In the studied company both the product and the process knowledge regarding interfaces was managed through so-called interface meetings which were considered as ineffective for this purpose in the component-oriented organisation of the studied company.

The method of engineering checklists is suitable for managing interface related product and process knowledge. There is however a lack of methods for creating and implementing engineering checklists which, though simple to use, may not be equally simple to create and implement. In relation to Research Question 2 it is concluded that the implementation of engineering checklists has to be supported with a dedicated method and it is important that this method caters for a proper contextualisation of the knowledge. This paper has proposed such a method and the evaluation shows that the choice of a visual tool inspired by Ishikawa diagrams is easy to use and provides results which are purposeful both for those creating checklists and for those using checklists. The main benefits of the proposed method are the simplicity of use and the fact that the designers themselves can use it without any need for a "knowledge engineer".

The evaluation however also indicates that there is a risk of user behaviour, also found in other knowledge management methods with the aim of knowledge codification, of producing specific solutions to specific problems without generalisation. This behaviour could render the resulting checklists to be perceived to be of limited value. This risk has to be addressed through clear instructions highlighting this issue and education of the users.

The planned future action for the proposed method is primarily to include the modelling of conversion mechanisms between inputs and outputs, just as sought for by one of the respondents. This might however bring about a large complexity into the method due to the iterative nature in which engineers work and the question is whether this increased complexity can bring about big enough benefits to justify its existence.

Acknowledgements

The authors would like to gratefully acknowledge the FFI Programme from the Swedish Governmental Agency for Innovation Systems (VINNOVA). We would also like to acknowledge the managers and designers for dedicating some of their valuable time for participating in studies constituting this research effort.

References

Adamsson, N. (2007) *Interdisciplinary Integration in Complex Product Development*, Doctoral thesis, **Royal Institute of Technology**, Stockholm, Sweden,

Almefelt, L. (2005) *Requirements-Driven Product Innovation - Methods and Tools Reflecting Industrial Needs*, Doctoral Thesis, **Chalmers University of Technology**, Göteborg, Sweden,

Almefelt, L., Andersson, F., Nilsson, P. and Malmqvist, J. (2006) *Requirement Management in Practice: Findings From an Empirical Study in the Automotive Industry*, **Research in Engineering Design**, Vol. 17, Nr. 3, pp. 113-134

Bathelt, J., Jönsson, A., Bacs, C., Dierssen, S. and Meier, M. (2005) *Applying the New VDI Design Guideline 2206 On Mechatronic Systems Controlled By a PLC*, **Proceedings of 15th**

International Conference on Engineering Design ICED '05, Melbourne, Australia pp. 415-416 (executive summary)

Bergsjö, D. (2007) *Management of Mechatronic Product Data in PLM Systems - Perspectives on Business and User Needs versus Technical Challenges*, Licentiate Thesis, **Chalmers University of Technology**, Göteborg,

Bergsjö, D. (2009) *Product Lifecycle Management – Architectural an Organisational Perspectives*, Doctoral Thesis, **Chalmers University of Technology**, Göteborg, Sweden,

Ćatić, A. and Malmqvist, J. (2010) *Requirements Management When Introducing New Mechatronic Sub-Systems - Managing The Knowledge Gaps*, **Proceedings of 11th International Design Conference - DESIGN 2010**, Dubrovnik, Croatia pp. 661-672

Clarkson, P. J. and Hamilton, J. R. (2000) *'Signposting': a parameter-driven task-based model of the design process,* **Research in Engineering Design**, Vol. 12, Nr. 1, pp. 18-38

Counsell, J., Porter, I., Dawson, D. and Duffy, M. (1999) *Scemebuilder: computer aided knowledge based design of mechatronic systems*, **Assembly Automation**, Vol. 19, Nr. 2, pp. 129-138

Dixon, N. (2000) *Common Knowledge: How Companies Thrive by Sharing What They Know*, **Harvard Business Press**, Boston, MA, USA

Fernie, S., Green, S. D., Weller, S. J. and Newcombe, R. (2003) *Knowledge sharing: context, confusion and controversy*, **International Journal of Project Management**, Vol. 21, Nr. 3, pp. 177-187

Huang, G. Q. (1996) *Design for X - Concurrent Engineering Imperatives*, **Chapman & Hall**, London, UK

Jansen, S. and Welp, E. G. (2007) *A Heterogeneous Modelling Approach for Domain Allocation in Mechatronics*, **Proceedings of 16th International Conference on Engineering Design ICED '07**, Paris, France pp. 707-708 (executive summary)

Kennedy, M., Harmon, K. and Minnock, E. (2008) *Ready, Set, Dominate: Implement Toyota's set-based learning for developing products and nobody can catch you*, **Oaklea Press**, Richmond, VA, USA

Kokkoniemi, J. (2002). *Checklists as a Tool for Collecting Experience Knowledge*, **IKS 2002, IASTED International Conference Information and Knowledge Sharing**, St. Thomas, Virgin Islands, USA,

Kokkoniemi, J. (2006). *Experiences from Generating Checklists*, **IASTED International Conference on Knowledge Sharing and Collaborative Engineering (KSCE2006)**, St. Thomas, Virgin Islands, USA,

Lundvall, B. and Johnson, B. (1994) *The learning economy*, **Journal of Industry Studies**, Vol. 1, Nr. 2, pp. 23-42

Morgan, J. M. and Liker, J. K. (2006) *The Toyota Product Development System*, **Productivity Press**, New York, USA

Munck-Ulfsfält, U., Falck, A.-C., Forsberg, A., Dahlin, C. and Eriksson, A. (2003) *Corporate ergonomics programme at Volvo Car Corporation*, **Applied Ergonomics**, Vol. 34, Nr. 1, pp. 17-22

NASA (2010) *NASA Lessons Learned Information System*, Retrieved on: 2010-08-05 from http://llis.nasa.gov/llis/search/home.jsp.

Pahl, G. and Beitz, W. (1996) *Engineering Design - A Systematic Approach*, **Springer-Verlag**, London, UK

Rahman, R. B. A., Pulm, U. and Stetter, R. (2007) *Systematic Mechatronic Design of a Piezo-Electric Brake*, **Proceedings of 16th International Conference on Engineering Design ICED '07**, Paris, France pp. 705-706 (executive summary)

Ribbens, J. (2000) *Simultaneous engineering for new product development: manufacturing applications*, **John Wiley & Sons**, New York, USA

Rinman, N. and Wilson, S. (2010) *Learning from Product Development Projects - An Approach to Increasing Product Development Capability Through Knowledge Creation adn Reuse*, Master Thesis, **Chalmers University of Technology**, Göteborg, Sweden, E2010:005

Schindler, M. and Eppler, M. J. (2003) *Harvesting project knowledge: a review of project learning methods and success factors* **International Journal of Project Management**, Vol. 21, Nr. 3, pp. 219-228

Sobek, D. K., Ward, A. C. and Liker, J. K. (1999) *Toyota's Principle of Set-Based Concurrent Engineering*, **Sloan Management Review**, Vol. 40, Nr. 2 pp. 67-83

Stokes, M. (2001) *Managing engineering knowledge: MOKA: methodology for knowledge based engineering applications*, **Professional Engineering Publishing Limited**, London, UK

Sunnersjö, S. (1994) *AID-Features: A New Tool for Design Automation*, **Proceedings of CACD94: Lancaster International Workshop on Engineering Design**, Lancaster University Engineering Design Centre, Lancaster, Great Britain pp. 241-258

Tischner, U., Schmincke, E., Rubik, F. and Prosler, M. (2000) *How to Do Ecodesign?: A Guide for Environmentally and Economically Sound Design*, **Art Books Intl Ltd**, Berlin, Germany

Wallace, K., Ahmed, S. and Bracewell, R. (2005) *Engineering knowledge management*, Design Process Improvement - A review of current practice. P. J. Clarkson and C. Eckert, **Springer Verlag London Limited**, London, UK, pp. 327-343

Ward, A. C. (2007) *Lean Product and Process Development*, **Lean Enterprise Institute**, Cambridge, MA, USA

VDI 2206 (2004) *Design Methodology for Mechatronical Systems*, **VDI-Verlag**, Beuth, Germany

VDI 2221 (1993) *VDI guideline 2221: "Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte"*, **VDI-Verlag**, Düsseldorf, Germany

Weber, M. and Weisbrod, J. (2003) *Requirements Engineering in Automotive Development: Experiences and Challenges*, **IEEE Software**, Vol. 20, Nr. 1, pp. 16-24

Vielhaber, M., Bergsjö, D. and Ćatić, A. (2010) *Mechatronic Systems Engineering - Theory and Automotive Practice*, **Proceedings of 11th International Design Conference - DESIGN 2010**, Dubrovnik, Croatia pp. 975-984

Williamson, K. (2002) *Research methods for students, academics and professionals: Information management and systems - 2nd edition*, **Centre for Information Studies**, Wagga Wagga, New South Wales, Australia

Ziemniak, P., Marek, S. and Stetter, R. (2009) *Mechatronics Engineering On the Example of an Innovative Production Vehicle*, **Proceedings of 17th International Conference on Engineering Design ICED '09**, Palo Alto, CA, USA pp.

CHALMERS UNIVERSITY OF TECHNOLOGY
SE-412 96 Gothenburg, Sweden
Telephone: +46 - (0)31 772 10 00
www.chalmers.se