Development of Industrial Information Systems based on Standards

Jonas Rosén

Doctoral Thesis
Production Engineering
Royal Institute of Technology
Stockholm, 2010
ABSTRACT

This thesis has studied how ISO STEP application protocols can be used as a base for development of enterprise information systems that provides information sharing capabilities in a collaborative environment.

In this thesis, five papers have been selected and are presented in the context of four themes that characterize the whole research body. The four themes are:

- Standard models as a base for development of IT systems
- Applications operating on integrated (standard) models
- Sharing of information
- Sharing of collaborative processes

The research have been performed in both an academic and industrial context, where the researcher have taken an active role in how to shape the solutions that became the result of the research project. Initially the projects were limited to virtual manufacturing with a focus on the management of manufacturing information. However, gradually the scope grew to include management of product information across the lifecycle. Also the organizational context was widened to include management of information that is shared between companies across their company borders.

The growing scope made the research to take on new aspects for each new area of issues that surfaced. One overall issue that surfaced was how to be able to exchange product data between companies over a long time and at the same time be able to impose configuration control of the shared information. One of the major conclusions is that sharing of information over time and across the lifecycle in a virtual enterprise is a task that involves more efforts than the task of exchanging product data of the same type (e.g. design data, CAD files) between two companies.

The results of the research show that the ISO 10303 application protocols are qualified for use in an enterprise product data management environment. However, the result also shows that there are a number of issues to deal with when developing a product data sharing environment based on standards. For example, how to efficiently transform and use the data structures of ISO 10303 application protocols in the different layers of a software system, the data layer, the business object and services layer and the user interfaces. Other issues are on how to deal with the neutral representation such that companies can integrate their contextual information with the neutral product data representation.

Future research is recommended to look more close into how the new evolving software architectures that becomes more mainstream today, such as Enterprise Service Bus technology (ESB) and Cloud Computing, amongst others, can utilize the ISO 10303 application protocols as their canonical format. Another important area is to further elaborate, and integrate ISO 10303 with existing technologies for Reference Data and unstructed data that today get more and more traction.
ACKNOWLEDGMENTS

This dissertation could not have been written without Professor Torsten Kjellberg who not only served as my supervisor but also encouraged and challenged me throughout my academic program.

My dear, older brother, Professor Bengt-Göran, who is to blame for inspiring me to take on the journey of research, deserves a very special gratitude.

Mattias Johansson, not only as Dr Johansson, but also as a friend who always knows what things matters the most.

Eurostep is a company like no other, where I have acquired an invaluable amount of knowledge through Torbjörn Holm. Håkan Kården who was brave enough to bring me on board. Per Brorson for always trusting me when the wind blow full storm.

All my colleagues throughout my research and work at Eurostep and KTH; Tobias Holm, Johan Nielsen, Magnus Färneland, Magnus N, Mats J, Matte, Martin, Staffan, Simon, Torkel, Olle, Samuel, Lankan, Jens, Peter, Micke, Per G, Örjan, Johan F, Väino, Fredrik N, Jonas V, Gunilla, Lasse W, Per J, you all deserves my gratitude for all the enjoyable moments together. Not to forget about Carmen…

Patrik Holmström and his family, always cheer laughter and cigars in.

I would like to acknowledge the financial contributors for the various (and numerous) research projects that comprised the research context and built up my industrial knowledge and provided me with powerful networks, Woxéncentrum and the Programme for Production Engineering Education and Research (PROPER). Also I would like to express my sincere gratitude to all of the companies that supported my research. Among others, Scania, and Volvo Car Corporation deserves a lion’s share of that gratitude.

To my mother, Mona, my brother Stefan, with his lovely family and Helena, my younger sister, always full of surprises. Also to my brother Bengt-Göran’s lovely family.

Lastly, I must from the deepest of my heart give all my love to the ones that occupies my heart at all times, Carl Duc and Maria.

….one is missing, Bengt. Dad, I really miss you.

Stockholm, May 2010

Jonas Rosén
THESIS STRUCTURE

This thesis is an aggregation of the appended five papers. This thesis comprises five chapters as described below:

The first chapter, the introduction, presents the research area, research thesis.

The second chapter, the frame of reference section, present the areas of concern that have been part of the research context.

The third chapter, the research and methodology, describes the author’s viewpoint of science and the research approach being used.

The forth chapter, the result, summarizes each paper and relates each paper to the overall research.

The fifth chapter, the discussion and conclusion, treats issues that are directly or indirectly connected to the problem description. Relevant issues to focus on in future research are also pointed out. The contributions are finally summarized in the conclusion.
LIST OF PAPERS

Results from the following research papers have been included in this work.

Paper A       A System Framework for Integrated Design and Manufacturing based on Standards
                J. Rosén, T. Kjellberg, and M. Johansson
                Proceedings of the CIRP 99 International Design Seminar, Twente,
                Netherlands (1999)

Paper B       Manufacturing Data Management based on an Open Information Model for Design of Manufacturing Systems and Processes
                J. Rosén, M. Johansson

Paper C       Collaborative Engineering, a tool for the Extended Enterprise based on an Open Information Model
                J. Rosén

Paper D       Federated Through-Life Support, Enabling Online Integration of Systems within the PLM domain
                J. Rosén

Paper E       Collaborative Change Management in the Virtual Enterprise, Enabling the Best Partner Vision
                J. Rosén, F. Almyren
DISTRIBUTION OF WORK

Paper A
Mattias Johansson provided the use case and data. Torsten Kjellberg contributed as a reviewer. Jonas Rosén designed the architecture and the framework.

Paper B
Mattias Johansson provided the use case and data. Jonas Rosén designed the architecture and the framework as well as all software prototypes.

Paper C
All work done by Jonas Rosén.

Paper D
All work done by Jonas Rosén.

Paper E
Fredrik Almyren provided the business environment, the use case and data. Jonas Rosén designed the solution architecture as well as the implementation of the solution.
OTHER PUBLICATIONS

Results from the following research papers have not been included in this work.

Journal papers
Johansson Mattias, Rosén Jonas: *Presenting a Core Model for a Virtual Manufacturing Framework*

Conference papers
Johansson Mattias, Rosén Jonas, Kjellberg Torsten: *Product Realisation and Virtual Manufacturing based on Standard Data Communication*

Johansson Mattias, Rosén Jonas: *Presenting a Core Model for a Virtual Manufacturing Framework*
CIRP 32nd Manufacturing Systems Seminar, May 24-26, 1999, Leuwen, Belgium

Rosén Jonas, Kjellberg Torsten: *A process-planning tool based on an open information model, integrating products, processes and manufacturing resources*
CIRP 2000 Design Seminar, May 16-18, 2000, Haifa, Israel

Rosén Jonas, Johansson Mattias: *Manufacturing Data Management based on an Open Information Model for Design of Manufacturing Systems and Processes*

Rosén Jonas: *Product Data Management for the Product Realization Processes in the Extended Enterprise*
SIG-Produkt Modeller, November 7-8, 2000, Linköping, Sweden

Wang Hongjun, Rosén Bengt-Göran, Rosén Jonas: *Inspection Information Modelling Based on STEP in Agile Manufacturing*
CIRP 2001 Design Seminar, June 6-8, 2001, Stockholm, Sweden
Rosén Jonas: *Developing Industrial Information Systems using Standards*

Rosén Jonas, Johansson Mattias: *Developing Information Systems for Sharing Product Data Through Life in the Extended Enterprise*
ISBN 91 631 5416 1

Rosén Jonas: *Web Services that enables on-line Product Lifecycle Support in the Extended Enterprise*
ISBN 91 631 5417 X
PDT Europe 2005, September 26-28, 2005, Amsterdam, the Netherlands

Rosén Jonas: *Enabling PLM using PLCS in a Service-Oriented Architecture*
ISBN 91 631 8854 6
PDT Europe 2006, October 16-18, 2006, Toulouse, France

Gulledge Thomas, Iyer Raj, Rosén Jonas, Hiroshige Scott, Johansson Mattias: *MODELING AN ENTERPRISE SERVICES ENABLED PRODUCT IMPROVEMENT PROCESS FOR MILITARY VEHICLES*
DETC2008-49922

**Other research publications**
Johansson Mattias, Rosén Jonas: *Informationssystem för virtuell tillverkning*
ISSN 1402-0718; 1998
Woxéncenter report

Roger Ljung, Johansson Mattias, Rosén Jonas: *Sharing Product Data Through Life in the Extended Enterprise*
# ACRONYMS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIAG</td>
<td>Automotive Industry Action Group</td>
</tr>
<tr>
<td>ALM</td>
<td>Application Lifecycle Management</td>
</tr>
<tr>
<td>AP</td>
<td>Application Protocol</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer Aided Design</td>
</tr>
<tr>
<td>CALS</td>
<td>Computer-aided Acquisition Logistics Support</td>
</tr>
<tr>
<td>CAM</td>
<td>Computer Aided Manufacturing</td>
</tr>
<tr>
<td>CAPP</td>
<td>Computer Aided Process Planning</td>
</tr>
<tr>
<td>CAX</td>
<td>Computer Aided Engineering/etc</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>ECM</td>
<td>Engineering Change Management</td>
</tr>
<tr>
<td>ECO</td>
<td>Engineering Change Order</td>
</tr>
<tr>
<td>ECR</td>
<td>Engineering Change Request</td>
</tr>
<tr>
<td>EDI</td>
<td>Electronic Data Interchange</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise Resource Planning</td>
</tr>
<tr>
<td>IDL</td>
<td>Interface Definition Language</td>
</tr>
<tr>
<td>MDA</td>
<td>Model Driven Architecture</td>
</tr>
<tr>
<td>MOF</td>
<td>Meta Object Facility</td>
</tr>
<tr>
<td>MRP</td>
<td>Material Requirement Planning</td>
</tr>
<tr>
<td>OLP</td>
<td>Off Line Programming</td>
</tr>
<tr>
<td>PDM</td>
<td>Product Data Management</td>
</tr>
<tr>
<td>PDT</td>
<td>Product Data Technology</td>
</tr>
<tr>
<td>PIM</td>
<td>Product Information Management</td>
</tr>
<tr>
<td>PLCS</td>
<td>Product Life Cycle Support</td>
</tr>
<tr>
<td>PLIB</td>
<td>Parts Library</td>
</tr>
<tr>
<td>SASIG</td>
<td>Strategic Automotive product data Standards Industry Group</td>
</tr>
<tr>
<td>SCM</td>
<td>Supply Chain Management OR Source Code Management</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>STEP</td>
<td>Standard for the Exchange of Product Model Data</td>
</tr>
<tr>
<td>UML</td>
<td>The Unified Modelling Language</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Service Description Language</td>
</tr>
<tr>
<td>XMI</td>
<td>XML Model Interchange</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Mark-up Language</td>
</tr>
</tbody>
</table>

xii
CONTENTS

Abstract
Acknowledgement
Thesis structure
List of papers
Distribution of work
Other publications
Acronyms

1 Introduction 1

1.1 Engineering information as a strategic resource throughout the product lifecycle 1

1.1.1 Geometry based product models 2
1.1.2 Product model and information model 5
1.1.3 An integrated product information model 8
1.1.4 Product information management 9
1.1.5 Collaborative information systems 10
1.1.6 Standards 11
1.1.7 Model driven development of information systems 16

1.2 Thesis statements 18
1.3 Scope and delimitations of research work 24

2 Frame of reference 26

2.1 The product realization process from an information system viewpoint 26

2.2 The Virtual Product, Prototype and the Digital Plant 33

2.2.1 Authoring tools 35

2.3 Enterprise Information Backbone 38

2.3.1 PDM 39
2.3.2 Application Lifecycle Management and PDM 42
2.3.3 Requirements Management and PDM 46
2.3.4 ERP and MRP 48
2.3.5 PLM 51
2.4  e-Business
   2.4.1 Electronic Data Exchange, EDI
   2.4.2 Trading portals
   2.4.3 RosettaNet
   2.4.4 Electronic Business Extensible Markup Language, ebXML
   2.4.5 Reference Data Libraries

3  Research and methodology
   3.1 Research context
   3.2 Viewpoint of science
   3.3 Research methods
      3.3.1 Information modelling
      3.3.2 Information modelling languages
      3.3.3 Component based development of digital information systems
      3.3.4 Model driven development of digital information systems
      3.3.5 Test cases

4  Results
   4.1 Paper A - A system Framework for Integrated Design and Manufacturing based on Standards
   4.2 Paper B – Manufacturing Data Management based on an Open Information Model for Design of Manufacturing Systems and Processes
   4.3 Paper C – Collaborative Engineering, a tool for the Extended Enterprise based on an Open Information Model
   4.4 Paper D – Federated Through-Life Support Enabling Online Integration of Systems within the PLM domain
   4.5 Paper E – Collaborative Change Management in the Virtual Enterprise Enabling the Best Partner Vision
5 Discussion and Conclusions

5.1 Evaluation of the results
   5.1.1 Standard models as base for development of IT systems
   5.1.2 Applications operating on integrated (standard) models
   5.1.3 Sharing of information
   5.1.4 Sharing of collaborative processes

5.2 Research validation

5.3 Conclusions

5.4 Future research

6 References

Appended Papers
   Paper A
   Paper B
   Paper C
   Paper D
   Paper E
1 INTRODUCTION

This chapter provides a background for the research, including research goals, scope, industrial and academic relevance.

1.1 Engineering information as a strategic resource throughout the product lifecycle

Today manufacturing companies need to be lean and agile to compete on the global market. During the eighties there was a strong focus on automation, “automate, or evaporate”\(^1\), whereas the keyword for the companies of today is “collaborate”\(^2\). These emerging trends in strategy that hits the surface continuously, they all somehow shapes the configuration of operational structure within and between companies and raise the level of awareness of how to orchestrate the business process and focus on core values within the companies. Nevertheless, to enhance the efficiency and quality within the corporate business processes, in spite of which new trend is shaping the operational structure, one fundamental driver is the ability to manage information and information flows.

Information, in general, is within this thesis used and defined as: (a) the information that is required to make decisions and take action throughout the whole life cycle of a product, the accompanying services and the delivered information, (b) information is structured into two types, static information (objects) and dynamic information (processes). Moreover, using information as a strategic resource means that the transformation/development/creation/usage of a real world physical object/artefact is supplemented with the information about the object. The information lives in a virtual environment, analogous to the real world object that lives in a physical world.

The emphasis on information management as one of the key drivers for successful product realization processes is identified by: (a) the Committee on Visionary Manufacturing Challenges for 2020 [VCM98] and (b) the Swedish national project Teknisk Framsyn [TekniskFramsyn00]. Specifically, amongst the Grand Challenges within the Visionary Manufacturing Challenges for 2020 **Information technology** is considered as a priority technology for all of the identified Grand Challenges, whereas another priority technology, related to

---

\(^1\) Automate, i.e. part of the lean manufacturing concepts such as Just In Time (JIT), Flexible Manufacturing Systems (FMS).

\(^2\) Collaborate, i.e. one of the key enablers of an agile enterprise [Kidd94].
information management in the context of this thesis, is **Collaboration software systems**. One of the five Grand Challenges is “Conversion of Information to Knowledge”; **“Effective conversion of information to useful knowledge in an environment where the volume of available information is increasing rapidly”**.

The Computer-aided Acquisition Logistics Support initiative (CALS) defines the use of sharing digitally supported product data as means to achieve efficiency for the overall business process [CTG97, Smith90]. The purpose of the CALS initiative is to enable enhancements in the business through: (a) improved productivity, (b) better quality and reliability, (c) lower costs, (d) shorter lead times. Further, CALS envisions some of the most important factors, in order to achieve their goals, to be: (a) re-use of information, (b) communication and transfer of information in an easy way, (c) usability and accessibility of information over time, (d) moving from a paper based methodology to digitally supported methodologies.

### 1.1.1 Geometry based product models

One way to improve the efficiency of information management is to handle information digitally, i.e. to use computer systems. Around the seventies and eighties Computer Aided Design (CAD) was starting to be used by industry to facilitate digital drawings, which further supported the product designers in their work c.f. Figure 1.

The first computer system that supported 2 dimensional drafting and was recognized as the first “CAD system”, was the SketchPad, invented by Ivan Sutherland at MIT. Soon after, vendors created commercial systems that were mostly just digital drawing-boards where it was as much effort to create a drawing digitally than as the former paper-based drafting, but much easier to edit. The 2D drafting was then enhanced by 2.5 dimensional drafting, which was used for CNC programming. 3D drafting was developed in generations, firstly having wireframe models, but they were hard to use in some respects, because of topological ambiguity. Surface modelling and solid modelling followed to more exactly represent all aspects of a real 3 dimensional object. Today, associative, parametric and feature based modelling puts a higher level of semantics to the object being modelled, but also starts focusing on moving the content of a 3 dimensional model to contain not just pure geometrical objects, but also product-oriented features such as material properties and behaviour.
Later, other computer system applications followed, like Computer Aided Manufacturing (CAM), Computer Aided Process Planning (CAPP), etc. These types of computer systems were the supporting tools that extended the engineers in their day-to-day work. Figure 2 shows how the product design activity produces CAD geometry (2D, or 3D) representations that are used by;

- (A) prototyping, e.g. Rapid Prototyping,
- (B) creating the tools necessary to manufacture,
- (C) CAM /CAPP systems to match designed product features into machining features,
- (D) manufacturing,
- (E) creating inspection programs, e.g. CMM programming,
- (F) documentation. Each activity, mostly, use other computer systems than the system being used to create the product model.

Figure 1. Historical overview of the CAD system evolution.
During the automation era it became evident that product information needed to be shared between the various activities within the product realization process. Product design models were being used downstream to prototyping, tooling and manufacturing preparation, but all these activities were using different computer applications with their own native formats. Hence, product information was digitally supported, but was hard to communicate and share. Consequently, standards were being addressed as means to bridge different formats of information between different computer systems. In the early days of computer aided engineering systems these standards, i.e. (a) Initial Graphics Exchange Specification (IGES) [IGES93], (b) Verband Der Automobilindustrie-Flachen-Schnittstella (VDA-FS) [VDA-FS86], (c) Standard Déchange et de Transfert (SET) [AFNOR85], (d) Data Exchange Format (DXF) [Autodesk88] and (e) ISO 10303-203 (AP203) [ISO203], were focused on geometry information. However, as more and more engineering information were created by other tools than geometry-oriented tools, the concept of product models were extended to cover all the information that comprised a product, such as properties, specifications, documents, process information, product structure, variants and meta-data. Hence, the information representation of products required more than geometry definitions. The use of “models”, in the context of product models, means a much more general information model than the former geometry model, which is from now on considered as one of the many constituents comprising the whole product model.

Figure 2. How product data originating from Product Design used downstream.
1.1.2 Product model and information model

There are many more aspects of information comprising the product model than geometry. When CAD systems were introduced and started to be used widely, the geometry, 2D, or 3D, was seen as the vehicle that carried the product model throughout many of the activities close to the product design. However, as the geometry product models were being communicated and transferred around, it soon turned out that there was a need to manage the access and administration of the geometry models, i.e. some sort of digital drawing archives and some sort of digital workflow. There was also a need to make more information about the product than the geometry available in digital formats, instead of paper based.

![Figure 3. Product Modelling in the Evolution of Product Development
Krause93. (Picture created by Johan Nielsen, KTH.)](image-url)

A product model is defined, according to the Special Interest Group for Product Models (SIGPM) as [Horkeby98] something that supports the product design
activity during the whole product development process. Further, each activity within the product development process works with a computer based model which is comprised by a number of different models, such as: (a) functional model, (b) organ model, (c) structural model, (d) design model and (e) production model, each related to one another. The important distinction between the traditional geometry based product model and the product model, is that the processes that use the product model are included as part of the product model and geometry models are themselves models comprising representations (and presentations) of the actual product model.

The SIGPM definition of a product model is by no means unique, but rather reflects a quite similar (or unified) view of product models that can be seen throughout the product design and product data community. The first proposal of ISO STEP reflects the early concepts of product information models by structuring information models that were built using building blocks (general resources) \[Owen93\]:

- (a) miscellaneous resources
- (b) shape; (1) geometry, (2) topology, (3) design shape, (4) solids
- (c) shape interface
- (d) product structure configuration management
- (e) material

\[3\] Functional model: the intended function, or set of functions that defines the capability, or behaviour, of the upcoming design solution.

\[4\] Organ model: the required functions needs to be realized in different phases within the design activity. The organ model is defined by Andreasen \[Andreasen92\] as something that represents a product concept in early design phases that later will be solved by product components. The ISO STEP community terminology defines the product component as being a technology concept that realizes product functions and which is later realized by different possible design items that are created in the detailed design activities.

\[5\] Structural model: each design entity, i.e. functional, organ (product component), item (part) and physical instance (serial numbered part), are usually structured into different views such as: decomposition (assembly-, spare-part-, maintenance-structures) and realization (function “A” is realized by product component “B” which is realized by part “123” and later manufactured as physical individuals “123”, “124” etc.).

\[6\] An engineering process that governs how product information is used is represented by a workflow. Engineering Change Management is an example of an engineering process that is represented within, or beside, a product model.
- (f) presentation
- (g) tolerances
- (h) features

A wider and more extensive definition of a Product Model can be found at Eurostep [Eurostep]:

“A Product Model stores all relevant information about a Product Model throughout the whole lifecycle in an integrated way: information that describes the product structure and its properties.”

This definition is wider in the sense that it does not origin from a CAD perspective and also does not have a design-oriented focus.

In this thesis the term Product Model means a model of a product. The definition of a model that is used within this thesis is:

“A model is an abstract simplified representation of a system or process, used to reduce the complexity of a problem.”

It is worth mentioning that a the widely used term Product Model is mostly used to capture some information about an existing product and create some sort of model of that particular product, i.e. a Product Model is a model that depicts an instantiated Product. In other words, a CAD model that represents the geometrical representation of a Product is also considered to be the Product Model. The representation of the information, i.e. how the information is structured together with the syntax of the data that constitute the content of the Product Model, is an information model. Conclusively, an Information Model can represent the content of a Product Model.

With the increased focus on the representation of the information comprising the product, the use of “information models and information modelling” was identified as an important part of the product modelling activity.
1.1.3 An integrated product information model

Two, or more, computer systems that can exchange Product Models between them needs to comply to the format of the information that is used to represent the Product Model. Consequently, if we want to re-use different kinds of Product Models for different computer systems where each system adds, or edits, information content that is specific for that system’s view of the Product Model, -then an Integrated Product Model is required. Hence, also an information model that manages to represent a wider scope of content is required.

As the consciousness of product models increased rapidly, efforts within various industry domains resulted in a myriad of different product information models, mostly with their own information modelling languages, scope of product information content and applications. The various industry segments created their own view of what was of importance for the information representation of a product model, e.g. mechanical engineering focused on geometry and mechanical assembly breakdowns, electrical engineering viewed the product from a functional perspective, the layout of circuits, the electrical properties, connectivity etc., oil and gas industry focused on the layout and configuration of process plants and petrochemical products, etc.. However, as product information needs to be exchanged across industry domains many of the different industry segments identified the need to harmonize their product information models. For this reason, the efforts within ISO STEP had a general view on product information models that is specialized with application protocols for specific industry segments. Thus, the importance of being able to harmonize different product information models is essential for efficient communication across industry and discipline domains. Nevertheless, in many ways the focus have been on representing the product model from a design lifecycle perspective. The major initiatives that address a (standardized) product information model from different views and using quite different techniques can be found in: (a) EPISTLE [ISO15926-2] and (b) ISO STEP [Kemmerer99]. Although the EPISTLE initiative to some extent is harmonised with ISO STEP, the actual product model concepts (or ontology) in EPISTLE are implemented in terms of a highly general information model, whereas the ISO STEP product models are more explicit in their nature.

The information that captures manufacturing processes and manufacturing systems need to be taken into account when utilizing communication within the product realization process. It is important that the manufacturing system and
the manufacturing processes can be represented on its own without being placed as secondary information attached to the product models. The manufacturing system can be seen as a mechatronic product and in many respects also be represented by sound product models according to Dr. Mattias Johansson [Johansson01]. If manufacturing systems and manufacturing processes can be represented adequately together with product models in one information model, in such a way that there are associations between each of them and with product specific information, we would have an information model that provides the means necessary for an integrated information model that enables concurrency on the level of information between the discipline domains of product (design and engineering), manufacturing (design, engineering) and processes (planning and manufacturing). On the other hand, having such an integrated information model would require a lot of effort to unify the different discipline domains, and today they each focus on their own affairs of state. In addition, each discipline has their own maturity in regards to management of information and information modelling in general. The product design domain has come a long way in the use of computer based information management, since they were the first to use computer-based systems. The downstream activities, i.e. manufacturing design, process planning and manufacturing, must act in accordance with the information that is received from the design department and thus the efforts have been on the exchange of geometry of product models.

1.1.4 Product information management

Management of product information have always been present at engineering and manufacturing companies. In the early days, drawing archives were being used together with administrative routines. Mainframes were being used later on to store drawings in printable formats together with in-house developed computer applications that registered part numbers and other administrative product information, i.e. meta-data.

The product information was mostly carried in and represented as documents and this started the development of Engineering Document Management systems (EDM), or simply Document Management systems. The document centric approach was later evolving into a slightly more product centric approach resulting in Product Data Management (PDM) systems during the eighties. These PDM systems provided functionality for workflow management, product meta-data, product structures and document-vaults.

However, PDM systems were not the only product information management systems available, but also other information systems were developed in the manufacturing domain, such as Material Requirement Planning (MRP),
Manufacturing Resource Planning (MRP II) and Enterprise Resource Planning (ERP). The MRP, MRPII and ERP information systems had another view and scope of information than PDM systems, but were also used as product information backbone systems.

Computer Aided Engineering tools such as CAD, CAM, CAPP, Finite Element Analysis tools etc., from now on considered as authoring tools, create the information that defines a product on the level closest to actual representation of a virtual, or physical product. PDM, MRP, MRP II and ERP systems manage and govern the workflow and the product definitions created by the authoring tools.

1.1.5 Collaborative information systems

The agile enterprise requires more than just being lean and flexible, according to Paul Kidd [Kidd94]. Being agile means to be able to build virtual enterprises based on each participating company’s core competences. Having companies work together in virtual enterprises means to work across the company borders in the day-to-day work. To enable an efficient Virtual enterprise the information base must be accessible by all partners in a secure way (some companies working together in one case can actually be competitors in another). The information flow in a Virtual enterprise implies the requirements of how to manage the information base and the format of how the information is accessed by each partner, as each partner have its own information systems and rules that governs the creation and manipulation of information.

Collaborative computer applications can either support asynchronous or synchronous collaboration. Asynchronous collaboration is here defined as how information between different companies is communicated between each company product information backbone system, e.g. company A is a supplier of parts to company B that needs frequent update of the parts supplied by company A. Company B imports the product information supplied by company A into its own PDM system and ERP system where each import updates and consolidates with the existing information base. Synchronous collaboration is here used when different companies’ information bases are updated in real

---

7 Virtual enterprise, companies work together to design, manufacture, deploy and service a product as being part of one and the same enterprise. The Virtual enterprise has a lifecycle and should be dynamically configured, e.g. companies can be contracted for a time and then be replaced by other companies later on.
A synchronous collaboration could be two or more companies working together with different CAD systems, at different geographical locations, to update a shared geometry model, but each CAD system is updated in real time and each update is maintained as a transaction. The synchronous collaboration can also be between the information bases in different product information management systems via real time transactions. Conversely, in order to provide information in a collaborative environment the information must be (a) accessible, (b) each participant must know what information to access, (c) there must be a possibility to have information communicated/transfered from various information sources and updated where appropriate, (d) information must be integrated, and consolidated\(^8\), both asynchronously and synchronously, (e) the information that is used during the collaboration must be managed and controlled and (f) the information must be in such a format that different computer applications can process it, technically and semantically.

### 1.1.6 Standards

When implementing agility, including collaboration it’s important that the computer based information is possible to share amongst each participating partner. Information is not just consumed, but also considered as a strategic resource, which means that it must also be possible to use in flexible ways. Analogous to the concept of being flexible in the context of the value creating processes, the information must be represented in such a way that it can be used and re-used in ways that are not known at present. The maximal information flexibility is thus achieved by having information being represented independently of how, and by what, it is being created and used within the value creating processes (the business processes). Further, by having the information being represented independent of how it is being created and used eliminates the need to re-structure the information when it is to be used by other kinds of value creating processes. One way to make information independent of the value creating process (how and by what the information is used and created) is to represent the information in an object-oriented way and standardize the information, c.f. Figure 4. The value of information makes it possible to improve the productivity in the value creating processes. To use information, to support and control the value creating processes, requires different levels of information flexibility.

---

\(^8\) Integration, in this context, means that information objects can be commonly shared and used by different systems. Consolidation means that information objects, being shared and used by different systems, are placed in context of each other.
In order to eliminate islands of information [Al-Timimi96] and point-to-point vendor specific and propriety solutions the solution is to use product data technology standards. These standards serve different purposes, such as (a) a standardized ontology of product and information models, (b) a standardized language (or languages) to represent the information models and (c) standard implementation of standardized information models c.f. Figure 5. An information model should not be dependent of which information modelling language is being used to represent it. If formal information languages are being used to represent information models, then it is possible to use implementation methods to automatically generate parts of a computer based information system.
The cornerstones of a computer based information system should provide flexibility for implementation of standardized product/information models so that the information base is not dependent of the systems being used (or rather the technology being used to implement the systems). Thus, information that has a longer life cycle than the technology, c.f. Figure 6, (software, hardware and operating system) can survive and be used without re-entering information (and therefore eliminate the risk of losing information and corporate knowledge).

*Figure 5 Three fundamental parts of a standardized information system development framework.*
Many different standards are on the arena of computer based information systems. Each standard, or standard framework, vary in scope and targets different subjects, i.e. information modelling languages, information models and implementation methods. Due to the increased use of Internet, many standards have emerged that focus on communication technology and the electronic trade processes; (a) eXtensible Mark-up Language (XML) \[\text{XML}\] is used to carry data in a neutral format, (b) ebXML \[\text{ebXML}\] and RosettaNet \[\text{RosettaNet}\] are examples of frameworks that defines the building blocks required to implement Business to Business (B2B), (c) the Unified Modelling Language (UML) is used to represent information models of all sorts, concrete and conceptual, in different phases of software systems development, especially to create state-of-the-art B2B systems based on, for example, Java technology. However, these days the focus on B2B means to support computer based information systems within the time-to-customer processes. In the area of time-to-market, there are already the Product Data Technology standards, i.e. the ISO STEP specifically, but they have had their focus, and impact, on the representation of information, independent of the computer-based systems.
used to create and use the information. Figure 7 shows how different standards are used for different things in a software system.

Figure 7. A layered structure that depicts different standards that are used by software systems. Each layer focuses on a specific application. Some standards can be used in other packages, e.g. the XML can be used as an information modelling language. The picture shows the most important standards.
1.1.7 Model driven development of information systems

The various standards for information modelling and information models, ISO STEP- and OMG [OMG] standards can each be used to create the various models for many of the development activities that constitute a computer based information system. However, each standard covers different aspects of the information system framework, c.f. Figure 8. The ISO STEP framework is strong on the information modelling methodology and the standardized information models, but the implementation method does not fully comply with state-of-the-art software development. The ISO STEP initial purpose was to provide information models that targeted the exchange of product model data and not information models that should be used to build the internal information model for a computer based information system. On the other hand, the OMG, and associated organizations, provides an information-modelling framework that supports state-of-the-art software development, but lacks useful standardized information models for the product data technology business domain.

Nowadays, the concept of model driven development is used frequently as a method to capture the information that specify the requirements of an information system to be built as well as the design and implementation of that particular information system. Figure 8 shows a model driven information system architecture that supports the use of information standards by:

- (a) Using Product Data Technology standards (ISO STEP) as reference models during the Business domain analysis and design.
- (b) Using UML [Booch99] to design the software system components and wrap business domain information entities into the software system design.
- (c) Use existing software code language bindings (Java J2EE, IDL CORBA, COM .NET [Java, J2EE, IDL, CORBA, IDL MS, COM, .NET] etc) that can be generated from the software system design models, (d) finally, compile the software system components and deploy them using software component standards (J2EE, CORBA, MS COM, MS .NET).
A model driven development approach separates design, implementation, coding and compilation. An example of a model driven development methodology framework is the Model Driven Architecture [Kleppe03], MDA, created by the OMG to facilitate the use of the UML for reusable component based development and automated coding based on modularised (layered) design models. The MDA framework use the terms Platform Independent Model (PIM) and Platform Specific Model (PSM).

The PIM and PSM are relative, for example, a truly conceptual model that describes a certain business domain, independently of how it is supposed to be implemented, is considered a (first class-) PIM, which then can be transformed into a data model, described independently of which storage mechanism to be used (relational, object-oriented, or file-based), which in turn is transformed into a physical database layout, a PSM, and later implemented in a vendor specific database.

Figure 8. A model driven information system architecture that supports the use of information standards.
1.2 Thesis statements

Thesis statement:

Standards driven through the Product Data Technology community can and shall be used together with standards originating from the software engineering and technology domain to provide the platform for Model Driven Development in order to build digital information systems that create and manipulate information the way its specified by the industrial domain of concern and to provide implementations that becomes the best of software technology at hand.

A computer based information system development process that is model driven will increase the cohesiveness between the stakeholder requirements, the information system design and the information system implementation. If the development process is not model driven the results that are created in one activity and used, as input to the next will gradually deteriorate for each consecutive development phase.

Figure 9 shows how non-model driven development of a computer based information system. Each development activity creates results that need to be translated and transformed into the next development activity. Every development activity is domain specific. Each activity use and produce information artefacts specific for their domain, discipline and with tools and formats mostly customized for their needs. This implies that traceability is difficult to establish in an efficient way. This is because if information is changed in one activity, for example requirements that are managed in word documents, then the other activities that depends on the requirements, for example the design of the database, or the code testers, then can only act if they are alerted by the ones doing the change. Also, the ones doing the change needs to know which things are impacted on beforehand.

A model driven methodology implies that each activity producing and using artefacts that comprise the software system, works with models that are shared and references each other such that changes in the information can automatically determine impact on the affected artefacts. Also, a model driven
development implies that various software system artefacts can be automatically generated from the models.

![Diagram of development phases](image)

**Figure 9. Non-model driven development.**

The thesis statement is constructed from a viewpoint originating from the software community in the sense of using as much as possible from the MDA [Kleppe03] and related standards. However, there is a strong issue inserted that points out the necessity to use Product Data Technology standards for the domain representation of information being used (conceptual information model). The emphasis in this thesis statement is on the collaboration and harmonized use of software-oriented standards and the Product Data Technology standards. The software-oriented standards should be used when creating the design\(^9\), implementation, deployment and maintenance models of the information system. The conceptual information models that can be found within the Product Data Technology standards will serve as the specification of

---

\(^9\) The design of the software information system, which in this case does not include the design of the conceptual information models, which otherwise is considered as an activity that is being part of the software system development process.
DEVELOPMENT OF INDUSTRIAL INFORMATION SYSTEMS BASED ON STANDARDS

the information that the information system should operate on and support the end users with, c.f. Figure 10.

**Figure 10.** Model-driven development process based on Product Data Technology and software engineering standards.

Supplementary thesis statement A:

**Standards are the key enablers for seamless information integration in a heterogeneous system environment where each system is governed by a certain business culture and goal.**

The standards in this statement refer to those being developed within the ISO STEP organization, the Application Protocols. Product data is represented throughout its lifecycle in the various Application Protocols. These Application Protocols provides full-scale PDM information models that targets conceptual
design, detailed design, manufacturing design, manufacturing preparation, design of product support, product support and maintenance. Since these standard Application Protocols have been developed and used by industry worldwide, there is an intellectual investment and computer based applications in place, which are important drivers. Companies also use standardized information models as reference models for their own corporate standards. More and more vendors of ERP, PDM, CAD, CAM, CAE systems also use standardized information models as references to their own. Another important driver that motivates the use of standardized information models is the fact that they reflect industry requirements. Moreover, to build Virtual Enterprises where each company needs to exchange information in a heterogeneous information system environment, standards is the only way to solve the problem of point-to-point solutions. Each company can have their own representation of information, but the information that needs to be exchanged must comply with available standards, otherwise new translators needs to be implemented for each pair of systems that need to exchange information.

Supplementary thesis statement B:

*Model driven development of digital information systems, based on standards, is a prerequisite for de-coupling information from the digital application systems used to create and manipulate the information.*

This thesis statement emphasizes on using standards to facilitate a model driven development approach. The model driven development concept is wider than the OMG Model Driven Architecture concept. The OMG MDA is narrow in the sense that UML is considered as the underpinning information modelling language. However, the model driven development concept is throughout this thesis used to extend the OMG MDA with those information modelling languages that today are outside the scope of OMG MDA, e.g. by including ISO EXPRESS. A model driven approach for development of information systems facilitates separation of conceptual information models (implementation independent) from concrete information models (design and implementation of the information system). This separation is important in order to maintain the integrity of the conceptual information model that is a model of a business domain. Otherwise, if there were no separation, the conceptual model would be influenced and designed in such a way that it suited the needs for the design and implementation of the information system. Thus,
the developed information system supports its users in their business accordingly to a conceptual model that is optimized, adapted and governed by the design and implementation of the information system.

Supplementary thesis statement C:

Successful application systems for Product Realization implementations requires both use of standards with the focus on software engineering and implementation technology, as well as domain specific standards that focus on information representation.

This thesis statement is constructed from a viewpoint originating from the product data technology community in the sense of using the STEP Application Protocols, in particular, and also other product data technology standards such as PLib [ISO13584] and EDI/ENGDAT [EDI, ENGDAT]. Product Data Technology information models have a much wider scope of content and surpass the information model functionality provided by for example vendors of PDM and ERP systems today. Further, the content of the Product Data Technology standards are truly implementation independent and does avoid software vendor lock-in. Moreover, there is a strong issue inserted that points out the necessity to use software standards for the digital information system design and implementation. The major Product Data Technology standard, ISO STEP, provides a complete framework for development of information systems, but the implementation methods are suited for building file-based data exchange applications, and not suited for creating a full blown large-scale distributed information system. Many of the ISO STEP Application Protocols, in particular the most used ones such as AP203 [ISOAP203], AP214 [ISOAP214], and what is believed to be the most used ones, AP239 [AP239] and the AP233 [SOAP233], have a semantically complex structure and are hard to understand if directly exposed to an end user. Thus, in order to effectively use one of those application protocols, one must add a layer, at least, that provides an end-user with the information in a way that one is familiar with. However, the semantic structure within an application protocol must be preserved, but exposed and presented in a simple and understandable way via a business-oriented layer. Figure 11 shows how an ISO application protocol defines the storage model of information. Due to its nature the storage
model have a high semantic complexity that needs to be hidden in the corresponding business object model that reflects the granularity provided by the originating end-user requirements.

Figure 11. Standards based architecture of a digital information system.
1.3 Scope and delimitations of research work

The model driven approach of developing digital information systems based on standards is certainly generally applicable, in spite of which industry domain and type of digital information system is being subject to development. However, this thesis narrows the scope by looking deeper into the digital information systems that support industries that design, manufacture and maintain electro-mechanical products in their operational environments. These kinds of companies, that have been the subjects of study for this research, are within the automotive and defence industry segment. One of the key drivers during this research, have been: Communication of engineering information within the extended enterprise, this driver puts the focus on collaborative digital information systems used for the exchange of engineering data within the time-to-market process.

The use of standards are limited to those frameworks that are state-of-the-art within product data technology and software engineering and that are considered as being the key contributors to the industrial environment amongst vendors as well as users. ISO STEP is the major contributor of product data technology and therefore also considered as the cornerstone for (integrated) product information modelling in this thesis. The Object Management Group, OMG, use a similar approach as the ISO STEP community, but their framework of standardization origins from the software systems engineering community and have another legacy than that of ISO STEP. The World Wide Web Consortia, W3C [W3C], contributes with widely used standards, mostly focused on Internet technology, but also includes other communities that standardize domain-oriented content and technology in regards to digital commerce and trade. All of the standardization bodies, ISO STEP, OMG and W3C, have significant overlaps and also in some areas works together, or use each other’s standards. ISO STEP includes OMG and W3C standards in their standardization framework. The OMG is in some of their standards harmonized with the ISO STEP, but this is mostly driven from the ISO STEP community.

Within this thesis ISO STEP is considered as the standardization body that contributes with the most relevant domain specific information models, as well as providing the most rigid and thorough information modelling philosophy. The ISO STEP provides implementation methods of high quality, including the use of OMG and W3C standards, but does not quite falls within state-of-the-art when used for developing state-of-the-art software systems with a state-of-the-art software engineering process. The OMG provides state-of-the-art standards
supporting the software engineering process in close collaboration with the W3C standards, but is in its infancy when it comes to the core values within information modelling and domain specific information models.

Figure 12. The ISO STEP, OMG and W3C standards are used in combination to provide a model driven development framework.
2 FRAME OF REFERENCE

This chapter describes the information systems used today and the requirements for an integrated product realization process from an information system viewpoint together with industry requirements of an integrated information model that is being developed within the ISO STEP community and other related Product Data communities.

2.1 The product realization process from an information system viewpoint

The product realization process includes development of products and the production system throughout the lifecycle. A product realization process includes a number of diverse activities that nowadays reaches, not only inside the enterprise boundaries, but to a significant degree also between enterprises. The co-operation between companies is often referred to as building an Extended Enterprise [Browne 99]. As competition demands shorter lead-time and increased performance as well as higher quality, this means that the key factors for the success of an Extended Enterprise needs to be focused on c.f. Figure 13.

Figure 13. The Product Realization Process.
Working in a cross enterprise product realization process means that the flow of information and material needs to be effectively managed throughout the entire life cycle, i.e. from concept to disposal. Many times, the collaboration imposes the need to work in parallel, or concurrently, i.e. according to Simultaneous Engineering (SE), or Concurrent Engineering (CE). Due to the topology of an Extended Enterprise, the flows of information and materials, becomes complex, especially if the configuration of an Extended Enterprise changes dynamically during collaboration projects. The Extended Enterprise paradigm also implies that OEMs, suppliers and customers will be involved in networks of Extended Enterprises. Figure 14 and Figure 15 shows how suppliers, customers, partners and OEMs participate in different projects, each with its own configured Extended Enterprise.

Figure 14. The Extended Enterprise (source Per Brorson, Eurostep Commercial Solutions AB)
The Extended Enterprise and the Virtual Enterprise relates to each other and the main target is the supply chain architecture. According to J.T. Mentzer et. Al. a supply chain is defined as [Mentzer01]:

“a set of three or more entities (organizations or individuals) directly involved in the upstream and downstream flows of products, services, finance, and/or information from a source to a customer”

J.T. Mentzer et. al. states that supply chains exists as phenomena, whether being managed, or not (for example as distribution channels). Supply Chain Management (SCM) is defined as [ibid]:

Figure 15. Network of Extended Enterprises (source Per Brorson, Eurostep Commercial Solutions AB).
“the systemic, strategic coordination of the traditional business functions and the tactics across these business functions within a particular company and across businesses within the supply chain, for the purpose of improving the long-term performance of the individual companies and the supply chain as a whole”

The Virtual Enterprise can be seen as a successor of the Extended Enterprise and extends the concept of dynamically configured business organization by including supply chains of information, knowledge and competence, as opposed to current Extended Enterprise paradigms that mostly focus on logistics supply chains. In this thesis a Virtual Enterprise, or Virtual Organization, goes beyond the scope of Supply Chain Management by defining an Virtual Enterprise as a Value Network according to Bill Gates statement [Gates99]:

“The old phrase supply chain implies links in a linear relationship, looking back from the retailer to distribution to transportation to manufacturing .... Today's approach is that of a 'value network'....Everyone who touches the product must add value, and communications go both forward as well as back. Companies in the value network aren't restricted to their places in line by heavy chains of process but can interact and do business with multiple vendors as they need to.”

In literature, that concerns product data and information management, the terms Extended Enterprise and the Virtual Enterprise are used interchangeably. Within this thesis the Extended Enterprise is considered as a type of a virtual organization, but with a scope on integrated supply chain of materials and each supply chain runs vertically within each life cycle phase of the product realization process. Whereas a Virtual Enterprise is considered as a true virtual organization that integrates supply chains of materials, data, information

---

10 The definition of logistics used here is: “the process of planning, implementing, and controlling the efficient, effective flow and storage of goods from point of origin to point of consumption for the purpose of conforming to customer requirements”.
knowledge and competence across the whole life cycle of the product realization process c.f. Figure 16.

Information management is one of the underpinning flows that enable efficient and effective supply chain management within supply chain organizations such as the Extended Enterprise and the Virtual Enterprise. Information integration, and consolidation, throughout the collaboration within, and across, supply chains includes: (a) accurate and accessible product information throughout the lifecycle of the product, (b) enterprise systems to coordinate resources, finance, logistics and operations, (c) supplier integration using online transactions, for example EDI, or other contemporary eBusiness technology (ebXML, RosettaNet etc.).
The supply chain of information requires information sharing capabilities that support the consolidation of the information of concern for the collaboration. Since information is produced and consumed throughout the life-cycle of the collaboration in different ways by different actors in the various segments by use of many different information systems, each with their own view and representation, there needs to be management of information, c.f. Figure 17 and Figure 18.

Figure 17. The supply chain of information.
DEVELOPMENT OF INDUSTRIAL INFORMATION SYSTEMS BASED ON STANDARDS

Figure 18. The supply chain of information, another view.
2.2 The Virtual Product, Prototype and the Digital Plant

Since the advent of digitalized drawings the use of computer based information systems has had a major impact on the engineering process in general. Firstly, the possibility of processing geometrical information in a computer lead to further engineering activities being computer supported, e.g. Finite Element Analysis (FEA), kinematical simulation, etc.. The computer based information systems being used by the engineer represented the physical product in a virtual world, i.e. the computer. Hence, as the virtual representation of the physical product continuously extended the amount of content that it could process for different purposes; new concepts were introduced that provided new ways of working in the engineering activities. Instead of having physical prototypes, Virtual Prototyping could be used to lower the cost and ramp-up for manufacturing. Virtual Prototyping is defined in many ways and closely related to other similar concepts, such as Digital Mock-Up, Virtual Reality and design-centered Virtual Manufacturing [Lawrence94]. Further reading about the Virtual Prototype and its definitions can be found in the paper “Definition and Review of Virtual Prototyping”, written by G. Gary Wang [Wang02].

The Virtual Product is a product design centred representation, but the extended use of virtual representations also encompasses the production of the virtual product. Accordingly to Lawrence Associates Inc., Virtual Manufacturing (VM) is the use of computer models and simulations of manufacturing processes to aid in the design and production of manufactured products [Lin95]. Further, Lawrence Associates Inc. defines Virtual Manufacturing as:

“Virtual Manufacturing is an integrated synthetic manufacturing environment exercised to enhance all levels of decision and control.”

The integrated representation of physical products, the manufacturing processes and the manufacturing system in a virtual environment is defined as the Digital Plant. Other synonyms for the Digital Plant are found amongst the commercial software system vendors, such as: “The Digital Factory”, “e-Manufacturing”, “E-Factory” etc. which can be further read about in the paper “Software tools for the digital factory” written by Danfang Chen, Torsten Kjellberg and Astrid von Euler [Chen09].
Computer Integrated Manufacturing [Bray98] (CIM) is also similar to the Digital Plant concept, but the CIM applications was mostly targeting the micro level of the Digital Plant, i.e. automating NC code and control system devices and Computer Aided Manufacturing (CAM). The Digital Plant concept is not only product design centered, but also looks on the manufacturing system from other views, such as: the production-, process-, preparation and planning, the manufacturing, the design of the manufacturing system etc.. Not only does the Digital Plant encompass the many different views, but more importantly, the product, process and manufacturing system over the whole life cycle. Figure 19 shows how the Digital plant encompasses the virtual product, process and manufacturing system and is connected to the physical world.

*Figure 19. The Digital Plant.*

The Digital plant can be seen as the control system for the physical plant, where the physical plant provides feedback to the Digital Plant that can self-adjust in response to the feedback. The Digital Plant includes not only the design of the product, process and manufacturing system, but also the manufacturing and operation of the manufacturing system.
2.2.1 Authoring tools

The information that defines a product is created by authoring tools such as CAD, CAM and standard desktop office tools; spreadsheet applications, word processors etc. Moreover, the authoring tools of today are not limited to capturing the product design in a digital format, opposed to paper-based drawings, but the tools themselves are part of the working processes in the sense that they implement certain algorithms and processing of the information being worked on. Hence, information being created by authoring tools is part of the corporate knowledge, as well as the authoring tool itself (if the information should be re-produced and given the fact that the whole set of information cannot be independently re-used by other authoring tools, also known as “vendor lock-in”).

Computer Aided Design, CAD, is used to create the geometrical representation of a product, or any system as such. Nowadays CAD systems manage not just geometry information, but also properties and behaviour of the system being modelled in the CAD system. As more features have been incorporated in the mature CAD systems of today, the amount of information grows and is also only available within the CAD system being used. The exchange of geometrical information to other systems is done by file-based data exchange using native format, or some ISO STEP (AP203, AP214). CAD systems are categorized as: (a) high-end, (b) mid-range and (c) low-end.

As CAD tools create the geometrical representation of a system, the behaviour is usually simulated and analysed by Computer Aided Engineering tools (CAE), which use the geometry definition as a base when applying computational algorithms to study the behaviour of the system under study. Different applications of computational algorithms are available to support tasks such as: (a) structural/stress analysis, (b) thermal analysis and (c) fluid dynamics.

As soon as there exists geometrical representations these can be analysed and visualized by other computer systems than the CAD systems. Visualization tools help downstream activities to perceive the look and feel of the computerized geometry product model by means of Virtual Reality and Digital Mock-up technology. In its simplest form visualization tool transforms the native CAD geometry into a lightweight format that allows the user to inspect the geometry model without changing the geometry model itself. More advanced visualization tools integrate technology that let many users share the same model in real-time in order to enable synchronous collaboration in a geographically dispersed environment, e.g. design reviews. Such visualization tools don’t change the underlying geometry, but allow users to create mark-ups
that are stored in the visualization model. Especially visualization tools supports many different native CAD formats and provides functionality to merge different CAD models into one visualization structure.

Part geometry and product structure being created by the design activities are published from the native CAD/product structure into visualization formats that enables other activities to analyse the product without the need of CAD systems, for example product review and marketing can give feedback and issue change requests. The published product structure can be viewed by lightweight CAD viewers, or by virtual reality technology. The important thing is to preserve the master product structure and associated geometry within the product model and have proposed changes being created as mark-ups in separate documents/files attached to change request information entities.

Within manufacturing preparation computerized tools exists that supports the generation of programming code for: (a) robots, Computer Aided Robotics
(CAR) and (b) machines, Computer Aided Manufacturing (CAM). CAR and CAM provides capabilities to simulate machine and tool paths in order to analyse manufacturability of the product as well as resource utilization for different possible ways to manufacture the product. Concepts such as feature recognition support automatic generation of machining operations given the geometry of the product. Thus the activity within manufacturing preparation of designing the machine, or robot programming code can be automated provided that the product geometry is accurate and sufficient for computerized analysis. Amongst the micro-level process, or operation preparation computer tools nowadays also Computer Aided Tolerancing (CAT) is recognized as an important system to feedback and analyse the impact of product design onto the manufacturing system capability. Figure 21 shows how CAM is used to develop and simulate a micro-process with given input from the product model and the resource model and the output is machine code.

![Figure 21. CAM.](image)

Whereas CAR and CAM works on a micro-level of the manufacturing system, other systems target the macro-level, such as: (a) Computer Aided Process Planning, (CAPP), (b) layout planning, (c) ergonomic design and (d) Discrete Event Simulation (DES). Process Planning creates a process plan on a macro level, encompassing product assembly and other processes that are required to manufacture the product. The process planning activity use input from the resource model and the product model and creates the process model and also
gives feedback to the product design and facility design activities c.f. Figure 22.

![Diagram of Process Planning]

**Figure 22. Process Planning.**

### 2.3 Enterprise Information Backbone

Product data management, PDM, is recognized as means to manage the product definitions being created by authoring tools, but also to govern the engineering workflow. PDM have its background in the early eighties and was then primarily used to manage digital drawings and other documents being created by the design departments. Nowadays the use of computers has spread to many of the other functional units within companies, not just the design department. However, other digital information systems exist that overlap the scope of PDM, i.e. Enterprise Resource Planning, ERP. ERP, from an engineering
perspective, provides information required by manufacturing of products and enterprise operations.

Most companies recognize the need for using backbone systems, such as PDM and ERP to make their business processes more efficient in terms of information management and utilization.

2.3.1 PDM

PDM manage product data throughout the whole lifecycle of the product\textsuperscript{11}. Hence, PDM is referred to as the “Product Information Backbone”. The backbone systems used today handles PDM using either a high-end commercial solution, such as: Windchill [PTC], Enovia [ENV], TeamCenter [Teamcenter], or by having an integrated solution where different legacy systems holds various functional parts of PDM information. These systems are used to store different views of part and product information often with cross references, but seldom using integrated information storage. Figure 23 shows a general view on legacy systems (based on studies of various companies such as Volvo Penta, and Volvo Cars amongst others).

\begin{figure}[h]
\centering
\begin{tikzpicture}
\node (1) at (0,0) {
\includegraphics[width=\textwidth]{legacy_systems.png}
};
\end{tikzpicture}
\caption{Legacy systems.}
\end{figure}

\textsuperscript{11} Product information being managed by a PDM system is mostly created only within the product design life cycle phase, but the product information is distributed and communicated to other life cycle phases.
DEVELOPMENT OF INDUSTRIAL INFORMATION SYSTEMS BASED ON STANDARDS

As the design domain was first to use computer-aided tools, the need to manage product information from these tools was early on targeted. The concept of management of product design information developed into the collective acronym PDM, Product Data Management. The concept of PDM have evolved into a number of different concepts, which more or less stands for the same thing, such as: (a) cPDM, Collaborative PDM, (b) EDM, Engineering Data Management, (c) PIM, Product Information Management, (d) PLM, Product Lifecycle Management, (e) VPDM, Virtual Product Data Management, (f) CPC, Collaborative Product Commerce. This is because different system vendors have labelled their products to emphasize on their prime features, and to differentiate in terms of marketing and the change of focus c.f. Figure 24. However, the trend is that most systems will converge to the PLM, Product Lifecycle Management c.f. chapter 2.3.5.

<table>
<thead>
<tr>
<th>Competitive Focus</th>
<th>1980s</th>
<th>1990s</th>
<th>2000s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Development Strategy</td>
<td>Margin</td>
<td>Market share</td>
<td>Market size</td>
</tr>
<tr>
<td>Technology Focus</td>
<td>Lower costs</td>
<td>Time to market</td>
<td>Innovation</td>
</tr>
<tr>
<td>Process Focus</td>
<td>Productivity</td>
<td>Data sharing</td>
<td>Leveraging intellectual capital</td>
</tr>
<tr>
<td>Organizational Focus</td>
<td>Serial design process</td>
<td>Concurrent engineering</td>
<td>Inter-enterprise collaboration</td>
</tr>
<tr>
<td></td>
<td>Departmental</td>
<td>Project teams</td>
<td>Agile market teams</td>
</tr>
</tbody>
</table>

Figure 24. The change of focus influences the usage and development of PDM systems and creates new acronyms for PDM-like systems (source Gartner Group and Accenture).

PDM systems, both legacy and commercial systems, have a focus on the detailed design phase of the product. More specifically, the focus of controlling the evolution of product design and the accompanying product data have by tradition, been comprised of; (a) management of parts, (b) management of product structures, (c) definition of manufacturing process, (d) variant management, (e) document management, (f) part- and product classification, (g) version management, (h) change management, including red-lining, (i) project- and program management. Each PDM project has its own specific implementation in regards to system architecture, process definitions
The trend today is that traditional PDM information needs to be extended to cover other parts of the product life cycle that have not been focused on before within PDM, such as:

- Early design phases: conceptual design and requirements handling, the requirement is to have full support for alternative design solutions and capture design intent together with the original requirements that control and defines the product behaviour. This part of product life cycle information is handled by standards such as AP233 [ISO233].

- Maintenance: the individuals manufactured together with information concerning maintenance activities such as anomalies and tasks. This part of product life cycle information is handled by standards such as Product Life Cycle Support (PLCS) [ISO239].

Table 1 outlines how different industry domains have different focus on what should be emphasized on for the product information.

<table>
<thead>
<tr>
<th>Product information focus</th>
<th>Industry domain</th>
<th>Drivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early design phase</td>
<td>Aero</td>
<td>Requirements on traceability from the part/product definition backwards to the product</td>
</tr>
</tbody>
</table>
DEVELOPMENT OF INDUSTRIAL INFORMATION SYSTEMS BASED ON STANDARDS

<table>
<thead>
<tr>
<th>Detailed design phases</th>
<th>Automotive</th>
<th>Part definitions, product structures, product variants, and product configuration.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product support, in-service support</td>
<td>Defence organizations, Process industry, off-shore, oil and gas</td>
<td>Design of maintenance and support tasks on designed products and physical individuals, work scheduling and follow-up on performed maintenance and support actions being executed on physical individuals.</td>
</tr>
</tbody>
</table>

Table 1. Different Product information focus.

2.3.2 Application Lifecycle Management and PDM

Mechatronic products developed today consist to a high degree of embedded software, which requires PDM systems to manage also the software artefacts alongside the traditional hardware.

Application Lifecycle Management (ALM) is nowadays the collective concept when looking on the lifecycle of (software-) systems development. Prior to the advent of ALM other concepts touched this from other angles, for example: Object Oriented design, Rational Unified Process [Kruchten99], Model Driven Architecture and some others.

Looking at software development from a historical viewpoint explains the rise of ALM. When developing software the formal artefacts managed by software was source code and eventually the deployed software. Other artefacts were more looked on as supplementary pieces, managed ad-hoc, for example software requirements. As software development matured, the business development identified the core tenants of a software development process based on the artefacts and activities that comprise software development; one example is the Rational Unified Process that emerged when harmonizing...
various modelling standards/notations into a comprehensive suite of design and development models, the Unified Modelling Language (UML). The driving factors success for UML was:

- (a) Introducing one modelling language that comprised the various design and development artefacts in an integrated modelling environment, the 4+1 view model, found in software development [Kruchten95].
- (b) Support in development tools for the modelling notation, for example round-trip engineering (model-> code->model).
- (c) Visualization and communication reaching outside the closed loops of software developers.

Historically, software to manage software development has primarily focused on the core artefacts that comprise the actual software, i.e. source code. The first integration in software development was to have source code tools integrated in the development environments. Later on modelling tools integrated with development tools, i.e. code generators. When modelling tools (e.g. UML modelling tools such as Rational Rose) became an integral part of software development, others followed by simply the fact that more things than just the core artefacts of software were being subject of information modelling. Thus, requirements started to become an integral part of the software model. These different artefacts usually were managed by its own software, thus integrations were done between the repositories of the different tools, thus no single unified data of the integrated information can be found.
Brittle repository-to-repository integrations are hard to establish and require much care and feeding.

Redundant, inconsistent functionality locked in practioner tools.

Figure 26. First generation ALM solutions [Schwaber06].
Also, it’s very difficult to exchange, or replace one part of the integration with another tool, since the integration as a whole is very brittle. For example, having another modelling tool to manage the development of UML models is very difficult if one suite of integrated products is used. Also, from a technical perspective, an integrated suite of software development tools is hardwired into the organization that uses it, for example, it’s difficult to let external developers in other companies to be part of such an environment.

Looking closer upon the current offerings provided by ALM vendors, IBM (formerly Telelogic, and Rational), Microsoft, Borland etc., the main chunks of information entities managed by the different tools can be broken down and categorized as:

- (a) Activity related information: The information mainly created by the Issue tracking tools and Configuration tools. This information is referencing the artefacts governed by the other tools that manage the artefacts comprising the actual software developed and deployed. For example; the Baseline
instances (found in CM), the Issue instances (found in Issue Tracking) are Activity type information entities, that have attributes/relations that reference other pieces of information.

-Artefacts comprising the actual software developed and deployed: The information entities mainly used to represent the artefacts created during design and development of the actual software. For example, UML models, source code files, Requirements objects, Requirement documents, etc.

Note, from an information perspective what is stored in each system's underlying data repository is a database representing some metadata in tables, but the actual content being in files, for example a source code tool managing metadata about the files that themselves are stored in a file structure (though the files and the file structure can be physically stored in a database). Requirements can be fine-grained and stored as metadata in a database, but document files can be also be considered as requirements.

Software Configuration Management (SCM) tools usually control software design artefacts whereas the deliverables, i.e. the compiled units, are not within scope for SCM tools. Integration of SCM and PDM has been recognized by industry as an emerging need [Persson01] and research have been conducted to exploit the area [Svensson03].

2.3.3 Requirements Management and PDM

Requirements Management (RM) is a discipline of its own with the objective to analyse and record requirements about a system. Today requirements engineering is carried out in its own domain in terms of information management and tools.

However, the early phase of requirements engineering creates information that strongly influence all other downstream activities within the lifecycle of the product. Requirements should be verified and connected to different information entities in the different design phases. When the product is manufactured and used, feedback is propagated backwards to the originating requirements, and then it should be possible to affect the design and enable users to validate any in-service modifications against the original requirements for the product. Hence traceability across the whole lifecycle of the product is necessary in order to track the impact of changes in both directions c.f. Figure 28.
Requirements management tools of today adhere to a common view of requirements that consists of: (a) recording requirements, (b) traceability between requirements (from/to), (c) text-based properties and (d) value-based properties. Requirements originate in many different forms, most often as requirements document, where the individual requirements are represented as sections or tables in the documents. Generally, requirements are specified as text, but also in diagrams, graphs, drawings, equations, numeric bounds, or tables of values. In order to use the requirements information it should be in a format that enables computers to interpret it. Having a formal representation of requirements enables other systems to integrate and consolidate the requirements information into the product model and thereby enables validation and verification outside the bounds of the requirements engineering domain.

In terms of integration between PDM and Requirements Management tools it may not always be obvious how to extract the requirements from the requirements document in a computer-sensible manner, but at least it should be possible to use the PDM document management capabilities to track the requirements documents. However, from an information management perspective the requirements entities and the product information should be configuration controlled based on for example verification and validation, or temporal conditions. This level of information integration between requirements information and product information would then enable consistency of the common and shared information base, but it implies integration and connectivity on the system level, so that different tools being used can create and update the information base according to common and agreed rules.

Seen from a PDM information perspective, requirements management tools are authoring tools that define the capabilities of a product which should be fulfilled by the design of the product, but from a system perspective, requirements management tools and PDM systems do not share any capabilities to integrate their respective information base, or any adequate data exchange, or API. Thus, integration of PDM and Requirements Management is done by placing requirements under the control of the document management feature of the PDM system, or by implementing attributes, in the product model in the PDM system, that reflects the requirements information.
Figure 28. Requirements affect the whole life cycle of the product model.

2.3.4 ERP and MRP

Enterprise Resource Planning (ERP) software is a set of applications that automate finance and human resource departments and help manufacturers handle activities such as order processing and production scheduling. ERP systems integrate enterprise functions such as sales, manufacturing, human resources, logistics and accounting. Within an enterprise an ERP system allows all applications to share a common database and business analysis tools.

The focus of an ERP system is to make the information flow integrated and consistent throughout the core business processes of an enterprise. As the targeted scope is wide and complex, in regards to the many activities that need to be integrated, this means that enterprise business processes needs to be reengineered and adopted to fit the implementation of the ERP system. Historically the ERP started out with functionality that targeted production planning, such as Material Requirement Planning (MRP I), but later evolved to include support for economy and finance activities and further on with personal administration.
MRP was introduced in the seventies with the scope of: material management, delivery estimates and optimal lot sizes. However, this was not sufficient to cover aspects such as: “available capacity”, “space”, and “capital”, neither did MRP I take into account the impact of released engineering changes and the cost associated with that. Hence, MRP I evolved into Manufacturing Resource Planning (MRP II) which further encompassed concepts such as: “long range planning”, “high level resource planning”, “master scheduling”, “rough cut capacity planning”, “detailed capacity planning” and “shop floor control”. The major difference between MRP I and MRP II was that a feedback loop from the actual production result to the planning activities was introduced to make it possible to enhance the processes.

Figure 29. The evolution of MRP to ERP. The overlap of PDM and ERP.

Companies with a focus on manufacturing needs MRP systems, whereas engineering companies, with the manufacturing being performed outside the company, needs PDM systems. ERP and PDM are fundamentally different in scope of functionality and view of the information being managed because they
each support different parts of the business process c.f. Figure 30. The Time-to-market view on the product model defines valid configurations of the product, which is used as input to the Time-to-customer view of the product model; however, the Time-to-customer view of the product model defines the as-built product model.

PDM is mainly responsible for managing product information that is created during product design. This product information is re-used by other users in other parts of the product lifecycle. ERP is used to manage, optimise and control processes within the company, which means that the information is valid during the time the processes are running.

![Figure 30. PDM-, ERP/MRP-focus (based on SAP AG).](image-url)
2.3.5 PLM

When vendors of PDM systems integrate early design phases and support for product individuals they say their systems are Product Life-cycle Management (PLM) systems. Steve Shoaf, IBM, states the following about PLM [Stark05]:

“PLM delivers application support for conceptual design, design engineering, manufacturing planning, service and maintenance. It is designed to neatly connect to and accommodate the workings of other processes, including:

- Customer Relationship Management (CRM)
- Supply Chain Management (SCM)
- Component Supplier Management (CSM)
- Enterprise Resource Planning (ERP)”

The general statement about PLM, provided by Steve Shoaf, illustrates a system-oriented view, which hides the details of information management. Actually the PLM definition cannot define the role by positioning existing system paradigms against, or beside, each other’s, since for example, a manufacturing-oriented company might implement PDM functionality within its chosen ERP system. Thus, in order to clearly define PLM we need to separate systems, business processes and information management.

Further, CIMdata defines PLM as [CIMdata02]:

“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, business systems, and information.”

CIMdata also states that PLM includes:

- Management of:
  - Product data and information—both its content and context
  - The design chain
  - Documents and their associated content (all types, formats and media)
Requirements (functional, performance, quality, cost, physical factors, interoperability, time, etc.)
- Product and project portfolios and product families
- Assets, e.g. plant machinery and facilities, production line equipment
- In-service information supporting service after sales

Program and project management
- Visualization and collaboration
- Component supplier management
- Digital manufacturing
- Product definition information authoring
- Product analysis, validation, and simulation
- Technical publications such as:
  - Service manuals
    - User guides
    - Assembly instructions

The PLM definition provided by CIMdata gives us another, but similar view on PLM, as Steve Shoaf, but it also adds some of the envisioned functionality, still on a system level. Neither of the PLM definitions presented by Steve Shoaf and CIMdata, have defined what is required in terms of information representation. Implicitly both of the PLM definitions state that a PLM environment requires integration between different business processes and different systems, both using different kinds of product models and other models. Thus, from an information representation perspective an integrated information model, enabling representation of different product models, and other models, is at least one of the pre-requisites within any PLM environment as such.

Moreover, the PLCS community defines PLM within the scope of the enterprise integration and the product lifecycle, c.f. Figure 31.
The PLCS initiative differs from the PLM vendor system perspective, by defining an integrated information model that supports PLM across the whole product life-cycle, independent from which systems, or business processes, that are used to create and work with the product information. Even though the PLCS targets the whole life-cycle, one of its key tenants are the parts of the information model that represents late life-cycle information, e.g. in-service support, integrated with earlier ones, such as the product design. The integration between product design and in-service support/maintenance is also taken on by the system-oriented community, which can be seen in the area of PDM and ERP integration nowadays c.f. Figure 32.


**Figure 32. PLM focus, integration of PDM, ERP and late life-cycle systems (based on SAP AG).**

### 2.4 e-Business

Supporting the logistics in terms of standardized electronic data exchange began with the standard for Electronic Data Interchange, EDI [EDI]. As many business process activities used computer systems to manage the information concerning the order flow (placing orders, shipment confirmation, pick-up at plant, confirmed on board, arrived at destination, goods received, inventory on hand), there was a need to facilitate an integration of the electronic information between the different parties involved in the order flow. EDI have been used since the 1970s by industry as the standard for electronic data exchange, but
today the EDI standard is only one of many other standards that have emerged to support digitally conducted trading between: (a) customers and commercial business companies (B2C), (b) between commercial business companies (B2B) and (c) commercial business companies and authorities/governmental organizations (B2A).

Since technology and business paradigms have evolved, new concepts for conducting commerce continuously surface. Amongst these are; (a) electronic market places (sell-side, buy-side and reversed auctioning), (b) Business-to-business collaboration, B2B and (c) on-line product catalogues. Although the e-business paradigm focuses on the time-to-customer process, there are many interfaces of information integration to the activities within the time-to-market process, both in terms of system and business processes, as well as information. Hence, to fully exploit the e-business paradigm it is necessary to provide an effective and efficient time-to-market process; otherwise the time-to-customer process will become a sub-optimised process.

Many standards are available, but the explosion of standards increases the complexity from a user perspective, i.e. (a) -Which standards should be used for which purpose? (b) -How does the implementation of a standard affects, or locks the selection of IT systems? And how current, or future, business process can be conducted when based, or being dependent, on standards. (c) -Is there interoperability between different standards? -Therefore a categorization of functionality, or usage, assists during assessment of the standards available for e-business. This categorization is mostly referred to as a stack, i.e. different standards are comprised of parts that each targets different layers in such a stack, similar to the Open System Interconnection model, OSI [ISO/OSI], c.f. Figure 33.
2.4.1 Electronic Data Exchange, EDI

EDI for Administration, Commerce and Transport (EDIFACT) and ANSI X.12 (the American equivalent for EDIFACT) have been the applications of EDI used for trading within Value Added Networks (VAN). Both the EDIFACT and the ANSI X.12 define a syntax and pre-defined documents and data-types that have been widely used by companies, some say that 90% of the larger companies have implemented the EDIFACT/ANSI X.12 standard [STA99].
An EDI document is referred to as a transaction set in the ANSI X12 standard. Some hundreds of different transaction sets are defined in the standard. The mostly used documents in an EDI system are: (a) purchase orders, (b) invoices and (c) related documents. The ANSI standard breaks up the various parts of standard documents into named segments and how these are to be used. EDI packages transaction sets for delivery over a network where each transaction set, or business document, goes into “departmental envelopes” called Functional Groups. A set of Functional Groups then go into larger “shipping envelopes” called Interchanges.

Figure 35 shows an example of how an invoice is referred to as a transaction set (Txn Set ID = 810) where all fields are segments, which further contains elements. A number of transaction sets are sent to a company and each transaction set is routed to an individual at the receiving company.
DEVELOPMENT OF INDUSTRIAL INFORMATION SYSTEMS BASED ON STANDARDS

Figure 35. EDI sample.

Figure 36. A Value Added Network, VAN, provides connectivity for EDI users.
A Value Added Network (VAN) provides communication of EDI documents between trading partners that have different hardware and software. To use a VAN simplifies the design of an EDI system for companies that cannot afford to handle the technical details of the communication, the connectivity, but instead can focus on the internal integration of EDI to the back-end systems, the syntax and semantic. The connectivity layer in the EDI stack is based on protocols such as: (a) X.25, (b) X.75 and (c) X.400. Facilitating a full-blown EDI system is costly, in terms of hardware and software, but having a robust, secure and reliable connectivity layer is a pre-requisite for automated business transactions. Further, a VAN provides built-in backup and audit capabilities in parallel with the key components service and support.

EDI as such does not include formalism to define the actual business process between trading partners. However, applications of EDI are governed by organizations that publish recommendations and usage guides. Strategic Automotive product data Standards Industry Group, SASIG [SASIG], is an example of such an organization. SASIG act as a virtual organization where each participating organization shares the chairmanship. The primary role for the SASIG is to provide product data exchange standards for the global automotive industry. One of its members Odette [Odette], provides the Engineering Data, ENGDAT [ENGDAT], standard which is an application of EDI for the exchange of CAD files. Hence the ENGDAT standard extends the use of EDI for activities within product development (time-to-market). One interesting issue is the use and definition of logistics within the SASIG and Odette organizations, differently from the traditional business transaction-oriented definition of logistics; the definition of logistics expands to include the whole life cycle of the product within the logistic supply-chain. Thus it can be expected to find EDI as one of the technology enablers within PLM.

Since the EDIFACT/X.12 standards explicitly defines a number of pre-defined business documents, which data-types to be used and the syntax, there is naturally a lack of flexibility in order to face the new evolving demands of how e-commerce is conducted with the exploding use of Internet technology, especially since XML is used as the language of the Internet nowadays. To accommodate for future needs Odette have initiated the Joint Automotive Data Model, JADM, to provide a XML Schema and UML model, via XMI, for EDI messages.

---

12 The members of SASIG are automotive organizations such as: Automotive Industry Action Group, AIAG (USA), Groupement pour l'Amélioration des Liaisons dans l'Industrie Automobile, GALIA (France), Japan Automotive Manufacturers Association, JAMA (Japan), Japan Automobile Parts Industry Association, JAPIA (Japan), Odette Sweden (Sweden) and Verband der Automobilindustrie, VDA (Germany).
2.4.2 Trading portals

Electronic Data Interchange has nowadays evolved to include electronic market sites using Internet standards. Moving from EDI-based technology, but still compliant with, the electronic market sites are implemented as portals. These portals syndicates electronic businesses such as: (a) procurement, (b) buy and sell, (c) auctioning, (d) reversed auctioning and (e) supply chain integration. From a logical and technical perspective, an electronic market site supersedes the role of the VAN in the EDI community.

Almost all initiatives that target electronic market sites rely on Internet standards and conform to the same architecture, c.f. Figure 33. Organizations such as CommerceOne [CommerceOne] provide both open standards and solutions that can be used to facilitate electronic market sites. Companies such as Covisint [Covisint] hosts market site solutions.

2.4.3 RosettaNet

RosettaNet is an independent, non-profitable, consortium whose aim is to enable partners to have seamless electronic business transactions in high tech supply chains – throughout an entire supply chain and on a global basis [RosettaNet]. A comprehensive framework has been established in order to solve business transaction issues targeting the supply chain by aligning electronic business (eBusiness) interfaces between supply chain partners.

One of the core components is the Partner Interface Process (PIP) specifications that normalize and specify the eBusiness process (transaction). The PIP is supported by an underlying implementation framework (RosettaNet Implementation Framework – RNIF) that specifies exchange protocols, and dictionaries that defines properties for products, partners, and transactions.

The intended use of RosettaNet is as an enabler for seamless electronic business transactions. As a result, RosettaNet specifies the processes needed to carry out a transaction, how to compose a message including attachments, the structure and content of the message (payload), and how to realize this in an implementation. The message content is, naturally, focused on business transaction related information such as quote and order entries and marketing information, but also includes more product related information such as bill-of-materials (BOMs) and approved manufacturing lists (AMLs).
RosettaNet provides a set of PIPs normalizing and specifying eBusiness processes including processes for the exchange of product information, service and support information, as well as manufacturing information. Each PIP specification includes a:

- Business Operational View (BOV) – captures the semantics of business data entities and their flow between roles in an eBusiness process.
- Functional Service View (FSV) – specifies the network component services and agents, and the interactions required to execute a PIP.
- Implementation Framework View (IFV) – specifies the network protocol message formats and communication requirements between supported peer-protocols.

These views define the specific sequence of steps required to execute a business process between supply-chain partners. The views also specify structure and content format of exchanged business documents (XML DTDs/Schemas\(^\text{13}\) and guidelines), as well as different constraints on these interactions, such as security and performance constraints.

A Business document format is specified either with an explicit interchange structure or with a reflective interchange structure. Both types make use of the RosettaNet dictionary to provide additional semantics to the content of a business document but in an explicit interchange structure the format syntax is provided by the XML DTD/Schema whereas in a reflective interchange structure all or part of the syntax is provided by a RosettaNet dictionary.

The RosettaNet Dictionary Architecture (RDA) provides architecture for the development of dictionaries and libraries. Dictionaries and libraries are instanced models of the dictionary and library schema of the RDA.

The dictionary (RosettaNet Dictionary – RND) schema represents metadata and is considered to be static/stable between multiple instances of transactions between supply-chain partners. As such the dictionary is the means to define and add additional meaning to instances of the generic entities of the library schema, and in some cases also the syntax. That is, the actual exchange data is represented in a library model and its meaning is defined by at least one dictionary model.

\(^{13}\) Both XML DTD and XML Schemas are used to define a PIP’s business document exchange format. The XML Schema provides a modular approach and business activities, actions, and attributes specification conforms to OASIS’ ebXML Business Process Specification Schema [Damodaran04].
The RosettaNet Implementation Framework (RNIF) provides the means required to execute business processes between two supply-chain partners by defining the overall RosettaNet business message format (MIME multipart/related and S/MIME multi part/related). The message format includes elements to support authentication, authorization, encryption, and non-repudiation. In addition, transfer protocol bindings as well as reliable exchange of messages specification are defined.

Figure 37 An example of a RosettaNet business message and container, and RosettaNet application protocol stack.
The RNIF is a container format which is used to carry all business documents as its payload. The container format is fixed for all exchanges, independent of PIP and business document type. It is also independent of the underlying transfer protocol used to transmit the message.

Figure 37 gives an example of a RosettaNet Business Message and underlying transfer protocols.

2.4.4 Electronic Business Extensible Markup Language, ebXML

ebXML [ebXML] is a holistic standard for the automation of electronic business. The standard has emerged through collaboration between UN/CEFACT and OASIS and is built on top of XML, to create an infrastructure for information-based and process-based electronic business. The standard addresses the different areas that constitute an electronic business process: (a) process, (b) trading partner agreement, (c) semantics, (d) notation, (e) security, (f) agreements, (g) standard information exchange and (h) standard information structure. Further, the standard is open and does not seem to compete with other standards within the e-business segment (but a significant overlap can be found with other process integration standards such as WFMC [WFMC] or BPMI [BPMI]).

2.4.5 Reference Data Libraries

Most companies have a need to manage corporate reference data, which can be found in for example, classification hierarchies, product classes, naming and identification schemas and product properties. Within the Extended Enterprise its vital to share common agreed information and make it accessible for use in a managed system environment during business transactions that spans across the company borders and systems. From an information perspective the concept of shared data can be within the range from data exchange between systems, and actually having a shared data environment, c.f. Figure 38. Nonetheless, when information is shared the context\(^\text{14}\) can be separated from the content\(^\text{15}\) in order to provide a more flexible information architecture not being hard-wired by implicit pre-defined contexts.

\(^{14}\) Context: shared assumptions
\(^{15}\) Content: the information that is exchanged
During the last decade, the requirements of having a standard representing common component libraries have resulted in the ISO 13584, also known as Parts Library, PLIB. The PLIB standard addresses the need to exchange information of pre-existing components. In 1990 the PLIB standard was launched at the ISO level and the goal was to develop a computer-interpretable representation of parts library data to enable a full digital information exchange between component suppliers and users [Pierra03].

Other initiatives have resulted in standards that targets Reference Data Libraries, or integration with, such as: (a) ISO 15926-4, the EPISTLE Reference Data Library [ISO15926-4] and (b) ISO 13399 [Nyqvist08].

Moreover, in some product information models, such as; (a) ISO 10303-214, (b) ISO 13399 and (c) ISO 10303-239, there are constructs that enables references to external reference data libraries. The reason for this is to increase flexibility such that maintaining and redundancy issues of reference data are kept outside the scope of the product information model.
Even though ISO standards target the issue of reference data libraries in a flexible way and with proper semantics it is not adopted outside the ISO community. Since reference data library is such a thing that resides in many places, in many systems and is used by many, there should be ways to accommodate for management of reference data libraries with mainstream technology in order to enable widespread use and in such a way that it can be integrated where being used by for example Product Information Management systems.

Today there exist technologies that can be used for reference data libraries within the W3C domain, such as the Web Ontology Language (OWL) [OWL04] that uses XML technology as the underlying transport mechanism. The OWL standard is part of the Semantic Web [Berners-Lee01, Daconta03] where the fundamental goal is to enable the use of the Web to provide information such that it is possible to process by computers rather than by humans. Thus a language that allows for more semantics than the HTML is required. OWL is defined by technology being derived from knowledge representation, logic and artificial intelligence.

Since the OWL can be used to represent the many things that already are defined within the ISO standards that targets reference data libraries, there is a strong opportunity to use OWL to represent ISO reference data libraries, hence reaching a wider audience and the possibility to use W3C technology for expanded functionality and facilitate computer system development based on mainstream technology. This is also an important issue that strengthens the role of ISO STEP standards, since all content that refers to reference data libraries can be facilitated by one technology. Further, separating reference data in an ISO STEP application protocol further strengthens the scope of the ISO STEP application protocol since the semantics of product data that is specific to industry can now be kept outside the ISO STEP application protocol, which focus on the semantics of how to represent product data.

An intermediate approach would be to expose existing reference data libraries with a standardized reference data library API using OWL as the representation. This would enable different PLM systems to have access to different sources of reference data. Hence, as different business processes use different PLM systems, mostly each with its own information representation, there would be a benefit if the common shared reference data were accessible, but governed by a reference data library system. Such a shared reference data library is responsible for controlling the reference data, thus minimizing redundancy and providing one point of access in the PLM system landscape.
3 RESEARCH AND METHODOLOGY

In this chapter the research methodology and research context are described.

3.1 Research context

The research has been performed in close collaboration with industry and academia throughout a number of different projects. Initially the research was carried out in an environment closely monitored by the industry called Woxéncentrum. Woxéncentrum was initiated by The Swedish National Board for Industrial and Technical Development (NUTEK) together with 27 other Competence Centres. The Woxéncentrum motto was: Efficient and Easily Changeable Production.

From 1999 the research was performed within the research Programme for Production Engineering Education and Research, PROPER, which was a long-term national effort to achieve excellence in areas of strategic importance for Sweden. PROPER has established cooperation between industry, the five Swedish technical universities, and the Industrial Research and Development corporation, IVF. Research projects have been performed in close cooperation between industry, universities, and the institute of production engineering research. The Swedish Foundation for Strategic Research provided the main funding for PROPER.

In 2000, the research was performed mainly together with the company Eurostep Commercial Solutions AB (today merged into Eurostep AB).

Figure 39 depicts an approximate timeline that relates the papers being used in this thesis, to when and within which research environment.
Digital Plant, PDMI2:

The research work being presented in this thesis has been carried out through a number of projects together with both industry and research programmes. During 1997 – 1999 the research was mainly carried out through Woxéncentrum. Specifically one project, PDMI2\(^\text{16}\) contributed to the major findings being used as a platform for the later phase of the research, c.f. Figure 40.

Researchers from KTH, Woxéncentrum, and Scania CV delivered: (a) guidelines and recommendations for applying AP214 as a core information model for manufacturing design, (b) a software environment for integrated manufacturing system design and simulation based on AP214 as the information backbone.

\(^{16}\) PDMI2, Product Data Management using International Standards.
**Development of Industrial Information Systems Based on Standards**

*Figure 40. The Digital Plant project within the international PDMI2 project.*

**Digital Plant, Prosolvia Systems AB:**

From 1997 until 1998 the research was performed partly with Prosolvia Systems AB in a European research project, ESPRIT project 25486, Interactive Evaluation Platform for Virtual Design and Production Planning (IEP). The IEP project goal was to deliver a simulation environment with a virtual reality user interface. The main research objective was to design the information infrastructure where end user tools such as PDM, CAD, discrete simulation and continuous simulation (Computer Aided Robotics) shared information.

Figure 41 shows how end users create edit and view information using CAD, PDM and simulation tools that are sharing collaborative engineering...
information. The shared information environment also binds together the simulation tools with virtual reality tools (not shown in this picture).

**End user systems (authoring tools)**

<table>
<thead>
<tr>
<th>CAD, PDM</th>
<th>Product Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAD, PDM, Electronic catalogues</td>
<td>Product Supplier</td>
</tr>
<tr>
<td>CAD, PDM, Electronic catalogues</td>
<td>Manufacturing System Supplier</td>
</tr>
</tbody>
</table>

**Figure 41. The IEP information infrastructure design.**

**Digital Plant, PROPER:**

From 1999 the research was performed through the Digital Plant project, which was an umbrella project within the research Programme for Production Engineering Education and Research, PROPER.

The overall research goals within the Digital Plant project was to: (a) investigate and demonstrate the usability of standardized information models shared as open data, (b) validate the STEP information models for use within manufacturing design, product modelling and process modelling.

The Digital Plant project has resulted in state-of-the-art studies, case studies and demonstrators within product and production development companies in
Sweden. Involved companies have been: Scania CV, Volvo Car Corporation, ABB Body In White, ABB Flexible Automation, ABB Corporate Research and Sandvik Coromant.

**Share-A-space, Eurostep Commercial Solutions AB:**

In March 2000 the achieved research results was implemented in the development of a commercial collaborative engineering tool, Share-A-space [CIMdata07]. The Share-A-space is developed at Eurostep Commercial Solutions AB. The Share-A-space software is based on the research findings that concern the model driven development.

The Share-A-space system provides functionality to manage Product Data throughout the design lifecycle phase and the in-support lifecycle phase. Share-A-space is further described by CIMdata [CIMdata07].
3.2 Viewpoint of science

The definition of science used in this thesis is based on the widely held common-sense view of science, accordingly to Alan Chalmers [Chalmers99]; “Science is derived from the facts”, “Science as knowledge derived from the facts of experience” and “Science is to be based on what we can see, hear and touch rather than on personal opinions or speculative imaginings”. Another, more general, definition can be found in dictionaries:

(a) “The observation, identification, description, experimental investigation, and theoretical explanation of phenomena.” [TAH06]

(b) “…systematized knowledge derived from observation, study, and experimentation carried on in order to determine the nature or principles of what is being studied” [Webster]

Furthermore, facts are assumed to be the basis for science, which lead to three claims, however more or less disputable, according to Alan Chalmers [ibid];

“(a) Facts are directly given to careful, unprejudiced observers via the senses. (b) Facts are prior to and independent of theory. (c) Facts constitutes a firm and reliable foundation for scientific knowledge.”

To most, the word science is actually a process, a way of examining the natural world and discovering truths about it. Conclusively, the essence of science is the scientific method\(^\text{17}\). Many schools look upon science, each having its own philosophy. It’s impossible to find a unified viewpoint of science, or a general theory for a scientific method, since some philosophies oppose each other’s epistemological definitions and research methods, and also claims others theories are wrong. Hence, as a researcher you must either chose the best of breed, or stick to the one most appropriate for the subject of research. The research comprising this thesis uses a best of breed approach.

\(^{17}\) The Supreme Court (U.S.A), in Daubert v. Merrell Dow Pharmaceuticals, Inc (509 U.S. 579), acknowledged the importance of defining science in terms of its methods: “Science is not an encyclopaedic body of knowledge about the universe. Instead, it represents a process for proposing and refining theoretical explanation about the world that are subject to further testing and refinement.” [RMSE00].
Research is usually categorized as qualitative, *-inductive*, or as quantitative, *-deductive*. Whereas the qualitative research is based from empirical results obtained from the real world that leads to (induces) theories, the quantitative research creates theories (hypothesis) about reality and verifies, or falsifies these theories when applying them. However, research can also be a mix of qualitative and quantitative, resulting in abduction [Alvesson94].

One of the most well-known philosophers of science, Karl Raimund Popper, argued that science rest on deduction rather than induction [Popper03]. Thus, accordingly to Popper, the general theory guides the researcher in making and testing specific predictions about future events. These deductions, i.e. predictions, are then empirically tested. The obtained result shows if the theory does, or does not support the predictions. Looking upon science with this viewpoint then induction is not needed, accordingly to Popper. Further, Popper claimed that the inferences that matter to science are refutations, which take a failed prediction as the premise and conclude that the theory behind the prediction is false. Moreover, a theory is a scientific theory if and only if there are possible observations that refute it, accordingly to Popper.

Thomas Samuel Kuhn, an American philosopher (1922-1996), argued that scientific practice is governed by a paradigm, a world-view sanctioned by the scientific community. A paradigm, according to Kuhn [Kuhn96], is a collection of procedures or ideas that instructs scientists what to believe and how to work. Kuhn argued that most scientists solve puzzles and problems whose solutions reinforce and extends the scope of the paradigm, rather than challenge it, also referred as normal science according to Kuhn. Some of these puzzles may become to hard to solve and are considered anomalies. If these anomalies start to build up, a crisis state is reached within that science, argued Kuhn. Then new candidates for a new paradigm may emerge to challenge the existing one. However, a new paradigm does not necessarily means that the old one is not true; merely the new paradigm solves puzzles better. Kuhn claims that a revolution takes place when paradigms are switched and that science is a progression of evolutions.

If a Popperian view on falsification is applied, i.e. failure to fit where grounds to reject a theory, then, according to Kuhn, all theories would be rejected at all times. For the purpose of this research, it’s important to consider different paradigms, as the thesis statement is founded on the use of two model driven development frameworks each following its own paradigm. The ISO STEP paradigm and the OMG MDA paradigm are used to create a new paradigm, but
not because the ISO STEP, or the OMG MDA paradigm is wrong, but because they do not provide feasible solutions for state-of-the-art development of backbone information systems for Product Data Technology solutions.

Underlying philosophy assumptions forms a particular research method. These underlying philosophy assumptions can be; (a) positivistic, (b) interpretive, or (c) critical.

The later part of the research have had a much heavier implementation-oriented approach, than the preceding research, which not easily fits into traditional scientific research since the implementation have been part of the actual research and not a result, or verification of the preceding research. However, looking inside the information system oriented research community reveals recognized scientific approaches for a research method, action research, where the researcher is actually actively participating in the case being subject for the research study, as opposed to traditional case based studies. According to Hult and Lennung [Hult78] action research can be describes as;

“Action research simultaneously assists in practical problem-solving and expands scientific knowledge, as well as enhances the competencies of the respective actors, being performed collaboratively in an immediate situation using data feedback in a cyclical process aiming at an increased understanding of a given social situation, primarily, applicable for the understanding of change processes in social systems and undertaken within a mutually acceptable ethical framework.”

Accordingly to action research, the laboratory is the real world, unlike for example, laboratory experiments, which struggle to maintain relevance to the real world. This is significant for this research where real world use-cases have been the subject for implementation during the ongoing research, i.e. industrial companies have been used when capturing real world requirements in the area of product data technology and these companies have also been using and giving feedback while using the implementation. The earlier research test cases where only using real world use-cases to capture product data technology requirements, which were then used for implementations that demonstrated possible solutions to verify the thesis statements.
3.3 Research methods

The chosen research method is in general following a quantitative approach where the theories are built upon the concepts of: (a) model driven development, (b) using product data technology standards, (c) using software oriented standards for the software engineering, and (d) component based development. However, an engineering science research approach is adapted upon the quantitative method. Instead of just analysing the real world and creating theories explaining the real word as it is at present, accordingly to Gunnar Sohlenius, an engineering science viewpoint is used to create theories that propose how the real world actually can look like, thereby suggesting new ways to develop new digital information systems. The research method also has a qualitative approach that, in parallel with the creation of theories, analyses the real world. The qualitative part of the research is more oriented towards the action research method as the participating researchers have influenced the real world use cases and the environment surrounding the use cases.

Following the engineering scientific approach [Sohlenius90] this research is structured and described according to the following processes:

(a) **Analyse what is**: - In this thesis this is the process that analyses the problem area and identifies the problem statements and requirements. This process is here strictly oriented towards analysis, i.e. in scientific terms directed towards a qualitative research approach. Chapters 2 and 4 specifically investigate and scrutinize current available resources that can be used as part of solutions.

(b) **Imagine what should be**: - This process is the synthesis, i.e. deduce a possible solution that met the problems found in process (a). This part of the research, in scientific terms, is the quantitative approach. However, the difference here, compared to natural sciences, is that the theory being proposed targets not only a solution that solves problems that are being found in a current state-of-affairs, but goes a bit further down the road to find a solution that implies new ways of working by providing new possible solutions to develop computer based information systems. The thesis statements in chapter 1.2 have represented the foundation for the methodology.

(c) **Create what has never been**: - This process implements, or materialize, the solution being developed in process (b). The results
presented in chapter 4 each present parts of the research performed based on the research thesis statements.

(d) **Analyze the result:** This process seeks to validate and verify the obtained results. Chapter 5 presents the concluding remarks, verification of the thesis statements and future research.

### 3.3.1 Information modelling

Information modelling is an activity (or the process) that creates, or edits, information models. An information model is defined in different ways, depending on scope and discipline, but a general applicable view on an information model defines it as being an abstraction of real world concepts [Rumbaugh91, Booch99]. Within this thesis an information model is considered to be a model (of information). A model, in general, is a set of statements about a certain system under study. Here, a statement is considered as an expression that can be true, or false. Hence, a model, or information model, is considered valid, or correct, if none of the expressions is false. Thus, if all of the expressions that comprise an information model are true, the information model is describing the system under study. Another way to use an information model is to let the information model be a specification for a system. Thus, a system under study is considered valid if all of the expressions comprising the information model are true. Furthermore, to precisely position an information model it should have a scope and a viewpoint.

This thesis covers two major topics concerning the use of information models. One of the topics, Product Data Technology, uses information models as representations of domain specific areas independent of implementation. The Product Data Technology viewpoint on information models is within this thesis considered to be *conceptual*. Whereas the other topic, design and implementation of digital information systems, is considered to have a viewpoint on information models that is *concrete*. The difference between conceptual and concrete information models is that a conceptual information model is not dependent on how it is being instantiated, or more precisely, implemented by a digital information system in terms of storage and programming language bindings [Schenck94].

The conceptual information model, as used within this thesis, defines the information base, how things are represented, stored and accessed by an information system. Things being subject of conceptual information modelling are considered being part of the *Universe of Discourse* (UoD).
"The universe of discourse is the aggregate of the individual objects which "exist," that is are independently side by side in the collection of experiences to which the deliverer and interpreter of a set of symbols have agreed to refer and to consider." [Peirce32]

The concrete information model takes care of how the information base is physically stored, the way that users manipulate the information base when interacting with the information system. Moreover, the concrete information model is used as a specification. Thus, the system under study is the computer system, or part of it, which is valid only if it conforms to the specification. Analogous to this is the fact that a conceptual information model is valid only if all statements are true for the system under study. Conclusively, since a conceptual information model leads to a number of concrete information models that specifies the implementation and the materialized computer based information system, then gives that the computer based information system must be model driven if it should be valid for a Universe of Discourse.

An information model is comprised by a set of models that each contributes to the understanding of real world concepts and for the design and understanding of information systems c.f. Table 2.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual information model</td>
<td>Used to capture the Universe of Discourse, to set the scope, a closed set. Defines the information base.</td>
</tr>
<tr>
<td>Concrete information model</td>
<td>Used to realize conceptual models for the purpose of storing and accessing the information base in an information system. The concrete information models capture the design and implementation of the information system that operates on the information base.</td>
</tr>
<tr>
<td>Data model</td>
<td>A concrete information model that is used to design the physical model (layout) that persists</td>
</tr>
</tbody>
</table>
the information base.

A logical data model can be used to describe the semantic relations between information entities, whereas a physical data model describes the layout of a storage.

| Activity model | An activity model describes the functions of an information system, or the activities and their relations in a business process. The activity model structures the activities and the information flows, the input, output, controlling mechanism and resources required by an activity. In many cases an activity model is designed before a conceptual information model in order to identify the information entities of concern within the given Universe of Discourse. An activity model is also useful to have as a representation of which activities a conceptual information model supports. |
| Workflow model /Process model | A workflow/process model defines a set of activities, their relations and the dynamics of a workflow/process in terms of temporal aspects such as when an activity is started, terminated, what triggers an activity, guarding conditions etc. |

Table 2. Different kinds of models that constitute an information model.

3.3.2 Information modelling languages

An information modelling language is a language used to express the statements in information models of a Universe of Discourse, or the system being the subject of study.
An information modelling language targets a certain kind of information model to be expressed. In general it can be said that an information modelling language is used to communicate the information model between different users, or systems. When it comes to selecting an appropriate information modelling language the decision depends of the usage of the information model, for example, should it be used to communicate the information between humans, between machines, or between humans and machines? The format, i.e. the representation of the information language also matters; -does it require the use of a lexical format, a visual presentation, or a combination of both? The nature of the content, the complexity of semantics and other factors also contributes to the choice of which information modelling language to use.

Within the ISO 10303 the information modelling language used is the ISO 10303-11 (EXPRESS) [ISO11]. The ISO 10303 application protocols are all formally defined using the EXPRESS language. Within the software engineering community, the Unified Modelling Language (UML) is used.

There also exist a standard that facilitates mapping between different information models. The MetaObject Facility (MOF) [MOF] is a language for defining languages e.g. UML. The MOF language is an OMG standard. Together with the XML Metadata Interchange (XMI) [XMI] MOF can be used to transform one information model to another.

In this thesis, the EXPRESS language is seen as a language that defines ISO 10303 application protocols. Thus, when implementing a Model driven approach the ISO 10303 application protocols are transformed into equivalent UML models. The semantic content of the actual information model is preserved in structure and content, such that it then can be used as a base for further transformations to different implementation models.

XML Schema [XSD] is within this thesis considered to be an information modelling language that is used for implementation models. Thus, XML Schema is not used when designing information models, it is an intermediate format used for defining implementation models such as Web Services WSDLs [WSDL], or as a file format used for XML content.

3.3.3 Component based development of digital information systems

A component based development approach has been used within this thesis in order to structure and manage the development of the digital information
system. An engineering information system resides in a heterogeneous environment, both from a technical and functional viewpoint, which therefore requires an architecture that provides and addresses a modular approach. The motivation for having a modular architecture is that it accommodates flexibility, plug and play behaviour and separates the functionality that components provide from how they implements it. Structuring a software system into modules [Parnas72] is a well-known approach and a fundamental principle of software architecture [Shaw95]. This modular architecture exists today and is referred to as component based architecture, Component Based Development (CBD), or Component Based Software Engineering (CBSE) within the software community.

A component, within the software community, can be defined as:

(a): "A coherent package of software artefacts that can be independently developed and delivered as a unit and that can be composed, unchanged, with other components to build something larger" [D’Souza98]

(b): "A Component is a unit of composition with contractually specified interfaces and explicit context dependencies only..." [Szyperski98]

The emphasis of the component definitions is the concepts of independent development of modules that are part of a whole, which can be reused and deployed for other future software packages, and that components conform to deliver services according to an interface that serves as the contract between the user of the component and the one who implements the component. Utilizing component-based technology also provides software architectures with flexibility with regards to programming languages and operating systems. The interface to a component can be implemented and consumed by other components without limitations to which programming language to use, if using contemporary technologies such as OMG CORBA, or Microsoft COM.

By using a component-based development paradigm it becomes possible to assemble new software systems built on existing components, or having legacy systems being used by new software if wrapping the legacy systems into components. This is a most important feature in today’s and tomorrow’s information systems environments where the number of software systems grow rapidly and the need for integration along with that. However, to successfully utilize re-use of components its evident that the component interfaces are designed in such a way that the provide services met the needed user
requirements being supported by the information systems. For example, common backbone information services, i.e. PDM, ERP, could be standardized, which means that future enterprise wide systems can have a plug-and-play architecture where different component-vendors provide their unique implementation. Hence, component-based development could decrease the problem of the vendor lock-in that exists today.

More specifically, a component is a unit of compilation that when replaced, or upgraded, does not need to force a re-compilation of the software systems that use the component. Further, the interface, or interfaces of a component defines the protocol of agreement between the user of the component and the provider of the component. Interfaces should be standardized in order to be widely re-used. Since the component implementation and the component interface are decoupled it enables software systems with a plug-and-play capability, hence, flexibility is achieved concerning run-time and deployment of components into installed software.

The protocol that a component provides for others to use is defined as an interface. An interface defines the data structures and methods that can be used when using a component. A component usually requires external services, required interfaces. An interface can be defined accordingly to different available Interface Definition Languages (IDL) such OMG IDL, Microsoft IDL (MIDL), or native programming language interface mechanism can be used, such as a C++ header file for traditional programming libraries.

An interface hides implementation from users using the component. A component should be replaceable by other components that can implement the same interface.

3.3.4 Model driven development of digital information systems

Information modelling have since long been used for the development of software systems. Mostly information modelling have been used as a way to provide layouts of the design of software systems in order to separate design issues from coding. Since the advent of the Unified Modelling Language (UML) there has been a focus on design of software systems by information modelling and implementation being generated from the design models. Within the software system development community the role of information modelling
have gone from documenting the design of software systems, to automatic code generation, mostly generating the stubbs, and finally to having information models that themselves can be executed and simulated while being implemented and compiled in the adjacent coding/build environment.

Code that is automatically generated (forward engineering c.f. Figure 42) is idiosyncratic, due to the optimization for targeted compilers. Thus, the traceability between the design model and the generated code, or another generated model, is violated. One way to address this issue is to formalize the forward engineering (code generation) with templates that explicitly specify the way a model can be customized when being used to generate another model [Selic03].

![Figure 42. Forward-, reverse- and re-engineering.](image)

Pertinent to a model driven development paradigm is to be able to experiment with a model to understand its behaviour before it is being implemented, hence moving the logical testing activity back to the design activity and provide concurrent engineering that affects the design and implementation discipline. The possibility to experiment with the information model is one of the key features of the Model Driven Architecture (MDA) [Kleppe03] and a language is standardized with this purpose, the Action Semantic language [Mellor02].

<table>
<thead>
<tr>
<th>Model driven concepts</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model execution</td>
<td>The possibility to simulate the behaviour of an</td>
</tr>
</tbody>
</table>
information model. Modelling environments that implement the Action semantic language can do this. However, these simulations mostly target the information models that specify the design of the actual computer system, i.e. the software application that implements the conceptual information models. The conceptual information models are to their nature specifying the static structure of the information that needs to be captured and populated by the software system. The conceptual model can be executed and simulated by means of creating populations, instantiation, which can be further analyzed and compared to real-world objects.

| Template mapping | When generating a new model, or code, for a model that is on a level further down in the abstraction layer, the transformation process should be specified as a mapping that allows for pockets that can be parameterized and customized. Thus an extension of mapping profiles can be built based on the re-use of those templates and, in theory, also automatically plug in into an existing model driven development framework. |
| Re-engineering | The round-trip of forward and reverse engineering. During model driven development each information model comprising the environment and being part of the forward and reverse engineering cycles needs to be synchronized across all iterations, or else the model driven development environment becomes inconsistent. |
| Reverse engineering | Once an information model, or code, have been generated it should be possible to update the information model that was used for the forward engineering with changes being done to the generated information model. |

82
Forward engineering  An information model should be possible to be used for the generation of another information model, e.g. transforming a conceptual information model into a concrete information model; the static structure is used to define the content of a database, the syntax of an exchange file, or the class structure of an object-oriented software application, e.g. Java, or C++.

<table>
<thead>
<tr>
<th>Table 3. Major concepts for model driven development.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object-orientation have been one of the key enablers for model driven development because a conceptual information model can be transferred into the design and development of the software system due to the fact that programming languages today are built with the same concepts as the object-oriented information languages. However, many of the different building blocks that comprise the software system are not necessarily object-oriented, e.g. the storage mechanisms, the data exchange mechanisms, or web-based clients using HTML and scripting technologies. However, such non-object-oriented building blocks can in most cases be represented in an object-oriented information model, e.g. relational database layouts are subject to object-oriented representations using certain UML profiles.</td>
</tr>
</tbody>
</table>

3.3.5 Test cases

The research have been performed in conjunction with industry in a number of research projects which have provided an environment for developing and verifying the research thesis, or parts of it. Since the research thesis comprises a general methodology based on the concepts of model driven development for a wide range of industry segments within automotive and defense industry, a step-by-step approach have been used when selecting the appropriate subjects for each test-case. One important part is the verification of using standardized information models, i.e. the STEP Application Protocols, to meet the requirements of industry. The STEP Application Protocols are more general in their definitions and covers a wider scope than industry requirements in general, which have led to substantial analysis work in order to identify which parts of the STEP Application Protocols to use and how to use them. Thus some test cases have focused on the analysis and applicability of the STEP Application Protocols, where the implemented application is merely demonstrating the proof-of-concept.
Another significant part of the research concerns the design and implementation of a computer system information system. The cornerstones provided by the two major standardization frameworks, i.e. ISO STEP and OMG/W3C, have each been subject for some of the test cases being presented. In the beginning of the research the ISO STEP framework was mostly used for the design and implementation of the early test cases mostly because the focus was on the analysis and verification of the content of the ISO STEP Application Protocols, in particular the ISO AP214. Later, the focus switched to find appropriate design and implementation frameworks that suited a general Product Data backbone information system based on new requirements that emphasized on collaborative engineering, supply chain management, and e-Business in the Virtual Enterprise. All presented test cases, except the ones concerning Share-A-space, focus on the use of ISO STEP implementation standards. The design and development of the Share-A-space software is the test case that addresses the combination of ISO STEP and the OMG/W3C standard frameworks.

The test cases have provided feedback to the research theories in an iterative manner. Since the test cases have been performed in parallel with the development of the research theories this have given the overall research work an abductive character.
4 Results

This chapter discusses the selected papers and how they have contributed to the result of the conducted research.

The selected papers each contribute in their own way to the overall research result. When looking in retro perspective there are four major themes that constitute the resulting research body. These themes closely correspond to the earlier presented research thesis and are derived from the statements found therein. The themes that outline the result of the research are: (a) Standard models as base for development of IT systems, (b) Applications operating on integrated (standard) models, (c) Sharing of information (including exchange of information) and (d) Sharing of collaborative processes.

The following sections describe each theme.

Standard models as base for development of IT systems: In particular engineering IT systems have been the subject of this thesis. Usually most engineering IT systems have front end (client) applications that create product definitions, like geometry and part characterization for CAD systems. The product definitional data is stored and managed by a back end system (server), normally a PDM system. The back end system is responsible for managing the product meta data, like identification, descriptive information and structural relationships, like for example configuration rules and Bill of Material (BoM) structures. This theme looks on the IT architecture aspects when developing these front-end and back-end systems.

IT architectures continue to evolve together with new technology. The main paradigm is distributed systems architecture, which then is shaped with new and evolving technology. Each new technology comes with new aspects concerning how to build IT systems. Throughout the timeline of this research the new technologies that have been developed clearly have moved the focus from being technology-oriented to becoming more business-oriented. Standards have been developed to support communication and transport of data in heterogeneous environments. The main technologies that have been studied throughout the research timeline are: (a) Common Object Request Broker Architecture (CORBA), (b) Service Oriented Architecture (SOA) and lastly (c) Cloud computing in the form of Software as a Service (SaaS) [Cloud].

85
Using standard models in this theme means how to utilize the product data technology standard models in software architectures using standard IT technology.

**Applications operating on integrated (standard) models:** This theme deals with how software applications can be developed using ISO STEP application protocols. Normally ISO STEP application protocols are used when exchanging product data between different systems and not as the foundation for an end-user application. Throughout the research different application domains have been studied when applying the ISO STEP application protocols. In the early research phases, the domain studied was the Virtual Manufacturing domain, because there was a need from industry to manage not only product design data, but also manufacturing resources and manufacturing processes in a similar way as with product design data. Manufacturing information is normally defined in CAD systems (drawings and models of tools, machines, factory layouts etc.), but at the early research phase there did not exist software that managed manufacturing information the way that PDM systems managed product data definitions. During the early research phase the ISO AP214 was the application protocol being evaluated for virtual manufacturing applications. Later on, the domain being studied was the collaborative supply chain. When looking at application requirements for the collaborative supply chain the lifecycle of product data appeared as an obvious target that needed to be addressed. During this research phase, the emerging ISO AP239 was used as a foundation for collaborative supply chain applications. Lastly, the scope of application has increased to include the systems engineering domain. Hence, the ISO AP233 is used for this last research phase.

**Sharing of information:** This theme has a prequel: “Exchange of information”, which is the initial scope for the early research phase where the product data was distributed to each application using ISO STEP files. Later on, the research came to include sharing of information as a key component in the information architecture. Sharing of information means that in order for different users, and systems, to operate on a valid set of information in a collaborative context, there needs to be a solution that manages the collaborative context and content across time and the lifecycle of the collaboration. The sharing of information solution makes it possible for different users, or systems, to access common agreed information at one single point, also product data is exchanged to this single access point, if not created directly in the access point. Sharing of information also means that consolidation of product data content is required as well as features that enable contextual information to be preserved and
referenced. Since different users, or systems, operate in their context, the content information in the common agreed area needs to be neutral, such that content can be used and manipulated in different contexts.

**Sharing of collaborative processes:** The concept of sharing information, such that it is neutral in relation to the different contexts used when being created and manipulated by different users, or systems, is the first achievement in establishing a collaborative information base. However, to further increase the responsiveness and awareness of change of information in a collaborative environment, information can also be communicated to and from different business processes. Since each partner in collaboration have their own private process, there needs to be a mechanism that can manage the inter-process communication in a neutral way such that each partner process does not become dependent on, or hardwired to, the other partner processes. The sharing of collaborative processes theme is exploiting the use of process integration by adapting standards for the representation of the content being communicated in the processes. Also by using a neutral coordination process, each participating partner only needs to communicate to one process. This theme came to live when looking closer to the concepts of process integration and when the concept of sharing information was in place solution-wise. It was first exploiting a distributed engineering change scenario, where a number of companies had their own private processes and needed to be coordinated. Later on, the coordination process as such was defined in a more general way and evaluated for use in other scenarios that were not strictly, or formally, following a change management process. Lastly, the research has looked on how to define a set of general coordination processes that can be parameterized and instantiated to fit into customized collaboration scenarios.
The figure below illustrates, approximately, how the selected papers relate to each theme time wise.

Figure 43: The research themes and how the selected papers are positioned.
4.1 Paper A - A system Framework for Integrated Design and Manufacturing based on Standards

Demands on shorter lead time imply more concurrent work and requires efficient communication of valid information. Not only do the demands on shorter lead time add to the complexity on an enterprise, the need to work efficient and concurrent with companies outside the corporate border in the extended enterprise also add to the complexity of managing the information flow. Today the development of IT technology and software systems address distributed computing, but one pain-point when working in a heterogeneous environment (heterogeneous business processes, systems and information) is how to be able to use and work with information independent of which system, or business process that was used to create it.

Looking on the product realization process, from a bird's eyes view, it includes systems that must provide users with information of many kinds, e.g. product data management (PDM), manufacturing simulation, computer aided robotics (CAR), CAD, CAE, as well as data produced from enterprise resource planning (ERP) and material resource planning (MRP). Each of these systems has their own view on how to represent the information of interest, which makes it difficult when for example developing integrations between different systems. The ISO 10303-STEP (Standard for the Exchange of Product model data) is an international standard for the computer interpretable representation and exchange of product model data, which addresses the previous mentioned problem concerning the data formats generated by various computer systems. The ISO 10303 STEP standard is a family of standards of which some can be used as a basis for implementing, sharing product model databases and archiving.

The use of ISO 10303 STEP standards for developing software systems is explored in this paper. This paper represents the initial research phase where the main workload was focused on understanding the role and usage of the ISO 10303 STEP standard when developing software systems (distributed architectures) using a three-tiered approach, see Figure 44.
In parallel with the study of the ISO 10303 STEP standard, various software technologies was tried out and evaluated for STEP data exchange conformance levels according to Timimi & MacKrell [Al-Timimi96], see Figure 45 and Figure 46. This including object-oriented database technology [ObjectStore], relational databases (using SQL adapters ODBC), CORBA using IONA Orbix [ORBIX], model-driven development using UML (Rational UML) to C++, STEP p21 read/write and STEP SDAI (using STEPtools [STEPTOOLS]).
Results related to Theme A:

This paper, and the research work being conducted at this point in time, focused on (software) technical aspects when looking into how to develop a computer system that access and manipulates product data defined by ISO STEP application protocols. At this point in time, contemporary software technologies for developing distributed software architectures were using a three-tiered approach using CORBA [CORBA], Microsoft DCOM [MSDCOM] and Java RMI [RMI] technology. When it came to Level 1 and Level 2 Conformance, according to Al-Timimi & MacKrell [Al-Tmimi96]...
STEP software tools were used from STEPTools [STEPTools] and EPM [EPM] to automatically create the data access and data transaction methods that comprise the Business Layer in a three-tiered architecture. The Presentation Layer was developed using Microsoft Visual Studio [MSVS] and Borland Delphi (which today is owned by Embarcadero Inc.) [Embarcadero]. For Level 3 Conformance, a model driven approach was elaborated. This approach looked into how to generate a CORBA interface based on the SDAI [ISO22], which would then be used to automatically generate client and server code for the Business Layer, using the IONA Orbix [ORBIX] software. The Business Layer code would then need to communicate to the Data layer using a database interface. The Data Layer was elaborated using an object-oriented database, the ObjectStore [ObjectStore] software together with code generation using the Rational Rose suite [IBM-RS]. Also, efforts were done to look into how to use SQL based databases as an implementation of the Data Layer. However, since relational databases introduce a mismatch, or impedance, to the object-oriented Business Layer, the SQL approach was not pursued. There was no end-user system developed within this research phase, instead knowledge and understanding of the possibilities when using STEP as an input for a model driven approach was brought forward as a base for the forthcoming research work.

**Results related to Theme B:**

From a research perspective, the novelty and contribution of how the ISO 10303-214 application protocol can be used to provide end-user functionality in software applications, is acknowledge to the co-author, Dr. Mattias Johansson. The decision to use AP214 as a base to support Virtual Manufacturing applications and not only as a data exchange format revealed a number of unthought-of issues concerning the development and design of the end-user graphical interfaces. Since the ISO 10303 application protocols defines the information representation of product data in a neutral context, there needs to be a transformation such that end-users easily can use and understand the information without knowing all the details of the very fine-grained information model. Hence, as a result, when applying an ISO 10303 application protocol as a base for an end-user graphical user interface, a number of business use cases should be defined that specify how the underlying information model should be populated and used such that it enables usability, hides the complexity of the information model constructs and can be used for the end-user business context.
4.2 Paper B – Manufacturing Data Management based on an Open Information Model for Design of Manufacturing Systems and Processes

This paper presents a solution that showcase how it is possible to use a standardized information model as a base for developing an environment where legacy systems as well as new systems could share information through use of a shared database. The scope of the application is design of manufacturing systems. The research work evaluated how the ISO 10303-214 application protocol could be used as an integrated information model for the various activities used for the design of a manufacturing system, such as manufacturing process planning, manufacturing system layout planning and simulation. In this part of the research the software applications are using the ISO 10303-214 as the conceptual model, which means that the graphical user interfaces contains functionality that directly operates on the information concepts in the ISO 10303-214 information model as they are defined by the standard. Normally software that is based on the ISO 10303 application protocols are developed as being translators, for example converting another systems information according to its native definition into the ISO 10303 application protocol in order to be exchanged to another computer system.

There was a software prototype developed that was comprised by two graphical user interfaces, the MDM toolbox and the Process Modeller, which were developed fully based on the ISO 10303-214 application protocol. These two applications created and manipulated information in a shared database. Also, the MDM toolbox was used to produce an export of manufacturing information that could be imported to commercial manufacturing simulation software, which in this paper is the Taylor Enterprise Dynamics (TED) product. Furthermore, during the research work, another simulation software (Simple++) [ARACOM] was used to showcase that the integrated manufacturing information actually could be used, and processed, with any other commercial simulation software that have a public API.

Thanks to the rich semantics of the ISO 10303-214 application protocol it was possible to explore different ways of how to create the input to the manufacturing simulation models. The information being created in the MDM toolbox was mainly focused on creating meta data about the manufacturing resources, in a hierarchical structure, representing a line with its cells and the resources comprising the cell, together with a spatial layout. The Process
Modeller mainly focused on creating meta data about a process plan and the breakdown of that into operations. The Process Modeller referenced information about product design and manufacturing resources. The input to the simulation model was created from the combined information provided by the MDM toolbox and the Process Modeller. Once the input information for the simulation tool was created it could be used in two different ways. One of those was to provide the simulation model with a manufacturing resource view, which showed how a manufacturing process could be simulated by how it was utilizing the manufacturing system from a spatial viewpoint, in order to investigate bottlenecks, throughput, manufacturing cell and line logistical arrangement. The other way was to logically look at the manufacturing process to analyse how to rearrange the order and sequencing of the operations, without looking on how the manufacturing resources logistically were used.
Results related to Theme A:

The research work for this paper is based on the evaluation of how the ISO 10303-214 application protocol could be used to support the design of a manufacturing system and how different computer system applications could work together on an integrated set of information. From software technical perspective the research work is based on the development of a STEP data exchange environment that conforms to a level 1 and level 2 according to Al-Timimi and MacKrell [Al-Tmimi96]. The research contributed to this theme by implementing software applications that used commercial STEP tool software, from STEPTools [STEPTools]. The simulation tool [ARACOM] was extended with an importer, being developed for this paper that was reading ISO 10303-21 files and translated that into (parameterized) simulation models that could be executed.
Results related to Theme B:

This paper is focusing on evaluating the ISO 10303-214 application protocol as the base for end-user computer applications that support the activities that comprise “design of manufacturing systems”. Two different, but integrated, use cases have been selected: (a) Manufacturing system layout and configuration and (b) Process planning. The paper mentions the Process planning use case briefly, but describes the Manufacturing system layout and configuration more in detail. These two use cases create and use product data that then can be processed and imported into a commercial simulation tool. The research successfully shows how the ISO 10303-214 application protocol provides a comprehensive information model to be used for implementing end-user use cases. However, the research also assessed that when using an ISO 10303 application protocol as a base for developing end-user applications, there needs to be a layer on top of the ISO 10303 application protocol that bridge expected end-user concepts to how the ISO 10303 application represents different information constructs. The reason for this is mainly that an ISO 10303 application protocol represents information entities and semantics using general concepts. For example, an end-user normally thinks of a manufacturing system to be comprised by manufacturing cells, lines and machines, in the ISO 10303-214 application protocol, the information entities that represents these end-user concepts are represented with information entities named product class, product component, item, item instance etc. Also, when matching the end-user concepts with the information entities described and specified by the ISO 10303 application protocol it turns out that many of the end-user concepts can be represented by different information entities with different constructs. The contribution from the research work, related to this paper, to Theme B in this respect, is that there needs to be usage guides and rules for how to populate and use an ISO 10303 application protocol for certain end-user use cases, otherwise it is most likely that information being created will be useless if it cannot be precisely interpreted.
4.3 Paper C – Collaborative Engineering, a tool for the Extended Enterprise based on an Open Information Model

This paper presents research that specifically addresses the concept of Collaborative Engineering in the Extended Enterprise. The research is based on industrial requirements on collaborative engineering that have been used as the input for commercial software. The development of the software is based on research about how to use ISO 10303 standards as a base for the software architecture and its end-user features.

In order to achieve collaboration between different partners in the Extended Enterprise, there needs to be common processes that support the distributed product development process. Requirements have been collected over the time from industry and the key requirements are described in the table below.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business contract</td>
<td>The exchange system, or collaborative environment, shall support the definition and maintenance of contracts regulating mutual information services between partners playing defined roles in the common business process.</td>
</tr>
<tr>
<td>Partner registration</td>
<td>Different partners play different roles in different projects. Each role defines which level of integration and interaction that is needed and allowed. Partners and roles are separated and the roles are set in regard to what level of value that is added to the business process.</td>
</tr>
<tr>
<td>Transparency of services</td>
<td>Users connected to a connected partner shall be able to navigate and use exposed information in the process owner PDM system. Specifically, the user shall be able to create and maintain subscriptions for distribution from the process owner.</td>
</tr>
<tr>
<td>PDM information hotel</td>
<td>The users in an extended enterprise shall have...</td>
</tr>
</tbody>
</table>
Table 4. Key requirements for extended enterprise collaboration.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export over time</td>
<td>The extended enterprise concept requires extensibility over time. For example, archiving of document issues (revisions, versions) will be affected by the extended enterprise requirements.</td>
</tr>
<tr>
<td>Project standards</td>
<td>Support is required for the publishing of different kinds of standards and specifications referred to in the contract between partners in the extended enterprise, e.g. design standards, standard parts, classification of parts, and preferred nomenclature.</td>
</tr>
<tr>
<td>Push, pull, transaction</td>
<td>Different modes of information exchange in the extended enterprise: broadcast (push to broad audience), distribution (push to pre-defined audience, derived, or subscribed), communication (person to person), pull if authorized, transaction (partner publishing information to virtual product, includes replication from partner to the process owner of selected elements of migrated structure).</td>
</tr>
<tr>
<td>Conferencing</td>
<td>Integration with services like video and audio conferencing.</td>
</tr>
<tr>
<td>Global workflow</td>
<td>The exchange system shall support multi-company workflow. A defined workflow shall be able to transfer itself to a partner’s support system.</td>
</tr>
<tr>
<td>Change management</td>
<td>When the common business process applies engineering change formalism, the corresponding messages must be in scope with the exchange of information in the extended enterprise and be supported by computer-based transactions.</td>
</tr>
</tbody>
</table>
When developing the software solution the ISO 10303 application protocols had two important purposes, on one hand, the software solution needs to be able to import and export product data according to a neutral format. On the other hand, since the software solution needs to manage product data that have been created in different systems in a neutral way, the ISO 10303 application protocol was used to define the various layers of the software; the database, the business object layer, and the graphical user interface, see Figure 48.

**Figure 48. A model driven approach.**

The initial approach for developing the software was to use the ISO 10303-214 application protocol as a basis for the software solution. However, while looking closer to the ISO 10303-214 information model and evaluating it, a number of issues surfaced. Amongst these issues that needed to be sorted out was the fact that the fine-grained information model was not always consistently modeled. For example, some information constructs were duplicated (instead of being reused), it was not always defined how to use some information constructs, and especially, since the ISO 10303 applications protocols are designed to work for exchange of product data, there was no
guidance, or rules for how to manage consolidation of different information sets that have an overlap.

**Results related to Theme A:**

This paper and the related research looked very close into how to completely develop computer software where all layers in the software are somehow defined by the semantics of ISO 10303-214 application protocol. The resulting software is based on a classical three-tiered architecture that operates in a web-based environment. At the point in time for this research, the contemporary software technologies addressed thin, or super-thin, web based clients, where an end-user only needs a simple web browser, like for example Microsoft Internet explorer, or Netscape. A large part of the research was spent on how to generate the persistent layer (a relational database, Oracle 8i) and the binding to that for the business-object layer. In order to serve as a truly logical three-tiered architecture, the user-interface calls on the business-object layer without knowing how the data storage layer works. The development approach was to preserve the object-oriented structure and let that automatically define the data storage physical layout. However, this was abandoned due to performance reasons and also because of development aspects. The findings of this research also showed that when using the ISO 10303-214 application protocol as the underlying information model in a model driven development approach, there needs to be some computer system infrastructure features modeled separately in order to superimpose on the resulting information model that generates the platform specific models for the various artefacts. The model driven approach helped to automate the definition of the data storage to a large extent, but not fully. However, the end-user interface layer and the business-object layer needed to implement most artefacts by hand.

**Results related to Theme B:**

The subject for the research was to develop a software solution for collaborative engineering based on the ISO 10303-214 application protocol. In this research collaborative engineering narrows down to asynchronous sharing of product data. When evaluating the ISO 10303-214 application protocol as the base of sharing information, as opposed to just exchanging information between two partners, it became clear that additional information constructs was needed in order to be able to consolidate information from many sources. The additional information constructs that were required mostly dealt with contextual representation. The ISO 10303-214 supports contextual information, but to a limited degree, mostly based on the fact that the applications using ISO
10303 application protocols use them for data exchange scenarios where one system exports product data which then is imported in another system. For example, many information pieces require some kind of identifier, and mostly these identifiers are governed by uniqueness rules. However, when sharing information, there needs to be a context defined for an identifier such that uniqueness rules can be executed based on the context of the identifier and not just the value of the identifier itself. For example, an identifier is valid for a certain organizational context, which implies that different organizations can have the same identifier values. Hence, an identifier in the context of sharing information is unique based on which identifier value in combination with the owning organization and which type of information it identifies. The changes required on the information representation was mainly adding contextual information constructs to the existing ISO 10303-214 application protocol together with some re-factoring of information constructs that were duplicated.

Most issues addressed previously, have their source in the fact that the proposed solution is based on sharing information, as opposed to exchange information. The focus on the applications using ISO 10303 application protocols is how to represent product data in a neutral file format such that it can be sent between different applications. What happens with the product data when it is being imported to, or exported from, an application is not within the scope of the ISO 10303 application protocols.

**Results related to Theme C:**

According to the requirements for collaborative engineering, that the research for this paper was based on, some fundamental features needed to be elaborated that had not before appeared in any known industrial application. The most fundamental feature is to enable collaborating partners to share information from their in-house systems with their partners such that pieces of the shared information can be put in context of each other. For this to happen there needs to be one place where the shared information can be managed. By introducing a separate software entity that manages the shared information, the concept of a Collaboration Hub was established. During the research the basic features of the Collaboration Hub evolved and was identified as: a) standard-based import and export between the Collaboration Hub and the source systems, b) product data consolidation capabilities, c) fine-granular security to preserve Intellectual Property Rights (IPR) for each participating partner, d) notification mechanism to be able to alert partners when information in the Collaboration Hub was changed, or updated, e) a Graphical User Interface to add and edit information in the Collaboration Hub.
Results related to Theme D:

The research at this time focused on establishing and elaborating the concepts of a Collaboration Hub, as explained in the section “Relation to Theme C”. In terms of process-oriented features, the product data that is consolidated and managed by the Collaboration Hub, does not involve any specific processes to govern how the product data is manipulated once managed by the Collaboration Hub. However, each participating system use processes governed by their business context in order to produce and use product data that is then exchanged to the Collaboration Hub. Thus, implicitly the Collaboration Hub needs to ensure that the consolidation of the product data can be executed such that the product data can be extracted and used in an appropriate way later on in the participating systems. Note, there is no explicit relation to theme D in this part of the research, but the awareness about coordination of processes in a heterogeneous business environment is raised.
4.4 Paper D – Federated Through-Life Support
Enabling Online Integration of Systems
within the PLM domain

This paper is introducing the ISO 10303-239 (PLCS) application protocol as an important resource for enabling information integration across the product lifecycle. Previous research used the ISO 10303-214 application protocol as the base for information integration. However, the lifecycle scope of ISO 10303-214 is limited to the design phase. Also, some important aspects when it comes to separation of product data into content and context are explicitly addressed by ISO 10303-239 and presented in this paper; see Figure 49 and Figure 50.

\[\text{Figure 49: Conceptual overview on how product content data and contextual data is separated, but linked.}\]
The paper also presents the draft version of the PLCS web services API. During this research phase the author participated in the European 6th framework project VIVACE, in which Service Oriented Architecture (SOA) [SOA1, SOA2] was considered an important piece of the puzzle when it comes to systems integration. Therefore, the PLCS standard was used as the base for developing a SOA web services API. The PLCS web service API was developed using a model driven approach, see Figure 51.

The PLCS web service API was also implemented as a part of the Share-A-space external API. Some client applications, that used the PLCS web service API, where developed and showcased as a successful result of the VIVACE project.

Figure 50: Example that depicts how PLCS is used when referencing contextual information.
Results related to Theme A:

This phase of the research focused on exploring contemporary technologies for distributed architectures. At this point in time the advent of the concept of Service-Oriented Architecture (SOA) influenced the product data community strongly. Until now, the product data standards provided implementation
protocols that addressed file-based data exchange, using the ISO 10303-21 and the ISO 10303-28 formats. An emergent need for developing implementation methods for web services resulted in the draft version of the PLCS PLM Services, which was developed as a result from work being done in the European 6th framework programme VIVACE. The development of the draft version of the PLCS PLM Services was done at the same time it was tested and implemented for a number of different systems (COTS) in the VIVACE project, as well as in industrial projects being executed at BAE Systems Hägglunds. In parallel the author was also exploring the OMG PLM Services [OMGPLMS07] standard and improvements was submitted in order to make the OMG PLM Services work in an interoperable manner between implementations based on the Java and Microsoft .NET platform.

**Results related to Theme B:**

The shared repository being used in Paper B was based on an extended version of the ISO 10303-214 application protocol. During the research comprising this paper, it was found that the scope of ISO 10303-214 was to narrow and therefore a major rework was done to provide the shared repository with capabilities to represent explicit late lifecycle information objects. The core model of the shared repository was merged with the ISO 10303-239 as a result of this. This proved to be successful and existing applications using the shared repository was not affected, since the shared repository was merging ISO 10303-239 and ISO 10303-214 and the overlap (design PDM information) was still intact.

**Results related to Theme C:**

During the research for this paper, the focus was on providing new software capabilities in terms of an online Web Service API, as explained earlier. Implicitly this resulted in that applications could more efficiently share information because product data could now be directly accessed instead of using file-based exchange (import/export).
This paper describes the implementation and use of an integrated change management process where participating partners each have their own private change management process. The research that contributed to this paper was evaluating cross company process integration from a technology perspective. When evaluating existing solutions and best practise for process integration using contemporary technologies like IBM Web Sphere, SAP NetWeaver (XI / PI) and Microsoft BizTalk, it was evident that the solutions (business orchestrations) normally place all the integration logic in a central place, which means that each integration becomes brittle and one-off solution.

This research looked onto how to let each participating partner, in a collaborative context, keep their internal process and let information at certain communication points be sent to a central shared process. This shared process is responsible for managing the routing (sending and receiving messages) between the different participating partners processes. This means that the work in the integration is done in each participating partners own process, and the central shared process act as a coordination process. The approach is possible because the central shared process also use a common shared information repository. Thus, a participating partner is doing work inside the local process accessing local information, but also have access to the common shared repository where links to other pieces of information can be used in order to more efficiently decide what information is to be updated in the actual collaboration.

An example that illustrates the process logistics when two private processes communicate via the engineering change management reference process is shown in Figure 52. In the example:

- (1) A partner starts an engineering change process by publishing an Engineering Change Request that has been released in the local system.
- (2) The collaborative process receives the “Register ECR/ECO”
- (3) which then is evaluated inside the Collaboration Hub before
(4, 5) a notification is sent to subscribing partners. The Collaboration Hub manages the state of the process and aggregates and consolidates incoming information with the shared information in the Collaboration Hub.

(6, 7) Next the notified partner receives the notification about the Registered ECR/ECO and determines what to do according to its private process and according to its role within the collaborative context.

(8) The notified partner needs to evaluate if the ECR/ECO is of interest, and if so, further queries about accompanying information to determine other characteristics about the ECR/ECO with inside the shared Collaboration Hub.

(9, 10, 11) When the notified partner request information about the ECR/ECO the Collaboration Hub filters out what information is allowed to be fetched from the notified partner. (12) Finally, the notified partner decides to create the ECR/ECO locally, which might later be reported back to the Collaboration Hub if it is promoted to an ECO for example.
The VDA4965 reference processes can be used as a tool to identify and map a partner’s private engineering change process into a collaborative scenario. By doing that, it becomes easier when more partners are involved in the collaboration since each private process only map to the reference process, also it simplifies things since there might only be a few activities inside a private process that needs to be communicated in a collaborative context. Moreover, having a reference process, like the VDA4965 [ECM06, ECR06, ECO], in place, it also neutralizes the information representation since the reference process defines the semantics of engineering change using the ISO 10303-214 (AP214). Hence, by using the reference process not only the process is managed, and controlled, but also which information should be exchanged, or shared. See Figure 53 for a schematic overview of the VDA 4965 ECR reference process.
This paper is based on one industrial use case where a part of the VDA 4965 standards were being applied. However, the research of coordination processes have been used in a number of different use cases, looking into a broader use of the coordination processes than just (design-oriented) Engineering change management for the automotive industry. For example, the coordination processes have been applied for more general processes like for example a highly distributed, and less formal, change management process that is initiated based on issues that occurs on product individuals, which are analyzed and eventually triggers a design-oriented change management process involving multi-disciplinary stakeholders. As an example of this, the coordination processes have been used for ECR and ECO to enable a product improvement process for military vehicles [Gulledge08], where a Condition Based Maintenance (CBM) process is using the ECR and ECO coordination processes. The integrated CBM and product improvement process for military vehicles was also looking at the DCOR reference processes to facilitate the integrated CBM and product improvement process. However, the ECR and ECO coordination processes that were elaborated during the research for this paper was easily adopted when it came to implement the integrated CBM and product improvement process for military vehicle business case, see Figure 54 and Figure 55.

**Figure 53:** Each partner involved in a collaboration maps their private processes into a shared collaborative process, here the VDA4965 ECR reference process is depicted.
Figure 54: Technical data and maintenance information (TDP= Technical Data Package, CM = Configuration Management) is updated as an effect of performing an ECO, which is part of the overall product improvement process depicted in Figure 55.

Figure 55: The Perform ECO as part of a product improvement process being based on the DCOR reference process [DCOR].

The shared collaborative process is technically defined as an orchestration where a number of receive ports and send ports are exposed as web services. These send and receive ports are the communication points between the participating systems and the shared collaborative process.
The VDA4965 reference processes [ECM06, ECR06, ECO] can be used to define which messages should be sent and received between the participants of the collaboration and the Collaboration Hub.

The Collaboration Hub architecture use standards to define the logical flow of the collaborative engineering change management process, by using the VDA4965; hence the messages being sent and received are identified by the VDA4965 ECM reference processes. However, the content is represented by the OMG PLM Services standard [OMGPLMS07]. Further the PLM information being stored and managed in the Collaboration Hub can be accessed using the OASIS PLCS PLM Services [PLCSPLMS07].

In Figure 56, the usage of the different PLM Web Services standards is depicted. However, since the PLM information used in the VDA4965 ECM reference processes have an overlap with the PLM information being represented by the PLCS standard, it would be possible to replace the OMG PLM Services with the OASIS PLCS PLM Services.

Figure 56: Collaboration Hub architecture.
Results related to Theme A:

During the elaboration phase of the research for this paper the focus, from a software architecture and implementation perspective was to exploit process integration technology as means to connect dispersed business processes and systems. One of the most fundamental building blocks, the message exchange mechanism was decided to be entirely based on an existing product data standard protocol. At the time of the research, the two primary candidates for the message exchange protocol was the OMG PLM Services 2.0 and the draft version of the PLCS PLM Services, since they both provided a Web Service API for product data. The initial approach was to only use the PLCS PLM Services since the chosen shared repository (Share-A-space) provided support for that out of the box, and also, the PLCS PLM Services provide a larger scope of information. However, the process integration engines (SAP XI and Microsoft BizTalk r6) being explored did not provide the full support required for using the fine-grained XML Schema that the PLCS PLM Services was based on. Thus, the OMG PLM Services was used to represent the messages being received and submitted to the shared repository via the process engine. The PLCS PLM Services was used internally by the process engine to get and manipulate the product data in the shared repository. The use of two different, but overlapping, standards implied an extra layer of mapping. However, the task to map and implement these two different standards was trivial, thanks to the semantic overlap.

Results related to Theme C:

More efficient private processes since they have access to a share information repository. Efficient for private process to work locally within their own business context and at the same time access referenced and related pieces of information through use of the shared information repository. This setup also increases the efficiency by how and when the communication between the private process and the shared repository should take place. The private process can decide to act upon changes on the information in the shared repository and the private process is therefore independent of the other participants since they all communicate to the shared repository and not directly to each other.

By combining the features elaborated concerning theme D with the concept of a shared repository, brought forward new possibilities for how to work with an integrated set of information with a lesser gap to the originating business context.
Results related to Theme D:

Earlier research, elaborated on how to use product data standards as means to provide a neutral view independent of the business processes used for creating and using the information, but in this research the goal was to extend the concept of “sharing of information” to “have shared processes”. More precisely, the concept of shared process in this research means integrating dispersed processes via a coordination process. By having a more process-oriented view, resulted in new possibilities of how to more efficiently use the shared repository when consolidating information. Without the process-driven interaction, information consolidation can be a cumbersome task to perform, since the quality of the information is depending on the capabilities the participating systems have for extracting information from their systems to import into the consolidating shared repository. For example, in most situations the participating systems extract a certain snapshot of their product data and import that into the shared repository, instead of just extracting a delta set of information. This implies that the shared repository needs to consolidate information that have already been processed earlier, but is slightly changed. Hence, to ensure consistency and quality of the shared information, the shared repository needs to investigate and analyze large portions of information already processed in order to find out how to consolidate in an appropriate fashion. By using a process-driven integration a participant can now instead identify certain events in his own system where it is of interest to exchange information to the shared repository, or conversely, identify change of information in the shared repository that should trigger an event that communicates via a coordination process to his, and others, system.
All intelligent thoughts have already been thought; what is necessary is only to think them again –

Johann Wolfgang von Goethe

5 DISCUSSION AND CONCLUSIONS

This chapter presents a summary and an evaluation of the research results. Further, the presented results are here validated against the research methodology and the thesis. Finally, the research contribution is accounted for.

5.1 Evaluation of the results

The results accounted for earlier, have been written over a long period of time, the first paper published 1998. Because of that this section will evaluate the results from a current timeframe perspective.

5.1.1 Standard models as base for development of IT systems

The research work assessed that there does exist building blocks that can be used for developing computer systems that operate on ISO STEP application protocols. Even though the ISO STEP standards themselves provide methods and specifications for how to implement data exchange according to the 4 Levels of Conformance stipulated by AL-Timimi and MacKrell [Al-Timimi96], a computer system cannot be automatically generated. Many of the interface and data persistence components in software architecture can be generated by the use of commercial STEP software tools, but there needs to be bindings and wrapper code implemented to make all the layers in a three-tiered architecture to stick together. Also, model driven development tools emerge outside the STEP community, but they are using other information modelling formats than what was used by ISO STEP standards at the time of research for Paper A. A best of breed development solution can be imposed which utilize model driven development tools from the software community which operates on the ISO 10303 application protocols, if they are transformed into an information modelling format, for example UML. An important observation is that the ISO 10303 application protocols provide high quality information models and as long as the semantics of those are preserved, the lexical format is of lesser importance. An important observation nowadays is that the STEP community has identified the need to get closer to the standards used by the software community, for example UML, OWL, REST, Web Services and XML Schema, BPEL amongst others.
At the time of research for Paper A and B, the concepts of distributed software systems heavily relied on CORBA and DCOM technologies. At the time being for the conducted research, the software prototypes that were developed mostly focused on file-based exchange and a shared database to prove the concepts of using a product data standard to define the data structures being used by dispersed software systems. Since Paper A and B was written, the software architectures have evolved into component-based and nowadays service-oriented paradigms. Paper D, which is written 2006, focuses on service-oriented architectures, and paper E further exploits the concept of service-oriented architecture for use in process integration. Paper A laid out an initial layered software architecture which still is relevant from a logical architecture perspective since it clearly positions where the product data standard model fits into the software architecture. Even today this has become more relevant since within the STEP-community there is feedback from users of the standards that require a layered architecture of the standards (eg. The OASIS DEX Library [DEXLib], AP214 usage guides) in order to simplify and make more consistent use of the standards when used to exchange product data. The layered software architecture presented in paper A, identifies this problem by defining the business object layer as responsible for how the information is actually stored in the database according to rules not defined in the standards, but rules that are evident because of the characteristics of the software system and the context which it is used in. Further, the stipulated business object layer also serves as a service layer where the complex information entities in the standard (which is represented in the data layout layer) are aggregated into less complicated information entities. Paper C further exploits the development of a three tiered architecture based on the ISO 10303-214 application protocol for sharing of information. The research behind Paper C is related to the implementation of the Share-A-space software product, where the model driven approach was used and evaluated in a software production environment.

5.1.2 Applications operating on integrated (standard) models

The results of paper B showed clearly that the ISO 10303-214 application protocol is suitable for representing an integrated manufacturing information set. By using contemporary software technology, it is possible to develop software based on ISO STEP standards. However, a side-effect was that when using an ISO 10303 application protocol for developing end-user functionality the fine-grained information model needs to be processed in order to provide an understandable context for the end-user. Further, using the ISO 10303-214 application protocol for manufacturing system development have been
investigated in detail and presented in other research work by Dr Mattias Johansson [Johansson01] and Dr Johan Nielsen [Nielsen03].

The research comprising this thesis has also looked into using the ISO 10303-239 in order to address more explicitly the late lifecycle of product data (paper D and E). Since the ISO 10303-239 was developed after the ISO 10303-214 some additional feedback was influencing the core constructs of the ISO 10303-239 such that Reference Data was playing an important role in order to allow extensibility. The Reference Data approach was decided to be used because then the ISO 10303-239 could be kept intact while externally defined Referenced Data could be developed and managed by end-users. The research work has been focused on how to implement the ISO 10303 application protocols themselves and not specifically on how to deal with the externally defined Reference Data. However, during the research work it has been identified that some of the ISO 10303 application protocols will overlap each other and one way to reduce the number of used standards would be to represent some of the standards as Reference Data. One example of this is the use of ISO 10303-239 to represent the full lifecycle of a system, or product, and use Reference Data externally defined, to represent explicit concepts that characterize the product, or system for different disciplines and domains. When looking back on the research, the findings of Paper B still makes sense, especially since the concept of developing computer systems based on standards is still not that common.

5.1.3 Sharing of information

In Paper C the concept of Sharing of information is one of the focus areas. The driver behind this concept was a set of requirements being gathered from industry by the company Eurostep. The requirement expressed the needs to be able to not just exchange product data between companies, but also to more interactively impose collaboration by use of a shared product data repository. This shared data repository has over time become labelled as the Collaboration Hub. Looking back on the research the main effort was put in the development of the software that materialized the Shared repository. The Shared repository was looked upon as a module that used ISO 10303 application protocols as the interface to the collaborative partners. At the time of the research (2001) the Internet was used primarily for consumer based applications (buy/sell commodities, travels, financial transactions) and there were not many or any at all, product data management applications at that time that utilized the web as the platform of implementation and communication. Therefore, much of the research effort was spent on development and evaluation of how to fully base a software on the, at that time, available web based software technology.
Nowadays, the web is the platform for most enterprise wide systems and core collaborative features have become commodities, for example portal application components, composite / mash-up applications, communication protocols that encompass security, transactions, master data management etc. However, still there are some missing pieces when it comes to how to actually represent and transform product data such that it can be used outside the enterprise for a wider audience. Moreover, the research focused on receiving and sending ISO 10303 product data in the collaboration, assuming the collaborative partners could provide their company specific product data in those formats. Today, also the concepts of the Semantic web and Reference Data Libraries come into play when companies want to externalize their product data. The combination of Reference Data Libraries and ISO 10303 application protocols is nowadays identified as an integral aspect when exchanging and sharing product data, but still this topic needs further elaboration since there exist different approaches how to solve this. Further, using Reference Data Libraries in combination with ISO 10303 application protocols but not just for data exchange and sharing, is further explored in academia and industry [Euler08].

Today, the ISO 10303 standards are accepted as means to exchange product data by industry, but the big hurdle to overcome is how to actually transform company specific product data into ISO 10303 product data such that it can be used and provide enough context and content for others participating in a collaborative environment. The acceptance of ISO 10303 by industry has also put the focus on which ISO 10303 applications should be used and as of today it seems that industries needs help to be able to mix different ISO 10303 applications protocols and use them for different purposes. Especially from an Enterprise Architecture perspective, the ISO 10303-239, 233 and 214 plays an important role since the overlap each other, but still each on their own contribute with specifics that are required. This is in many cases a problem since companies’ wants one unified information model as part of the PLM backbone.

At the time of the research, the Shared repository was established as some sort of a hosted service that was accessed as an Internet application. Today, the concept of hosted services has evolved into things such as Cloud computing [Cloud] which encompass Infrastructure-as-a-Service, Platform-as-a-Service and Software-as-a-Service. The Cloud Computing aspects are of most relevance to the Shared repository and something which needs to be addressed
in order to further position the Shared repository as a vital component in today’s collaborative environment.

5.1.4 Sharing of collaborative processes

The concept of *Sharing of collaborative processes* is derived from the concept *Sharing of information*. In Paper E, the first real implementation approach was established initially based on the German VDA 4965 ECM standard. However, throughout the research work it became evident that the studied collaborative process could easily be formed in a more general way in order to provide flexibility in order to target a wider audience than just the ECM processes. The VDA 4965 ECM processes are a direct result from the German automotive industry targeting Engineering Change Management, and with the result being a reference process that automotive companies should use when mapping their own business context specific Engineering Change Management processes in collaboration with other partners, or OEMS. The scope addressed with the VDA 4965 is very much design-oriented, typical mechanical and manufacturing design. However, the scope of the VDA 4965 was found to be quite narrow, because when partners and OEMS have the possibility to connect their internal process, it also automatically enables other domains and lifecycle stakeholders to participate such as systems engineering/requirements management, in-life/support and maintenance. As a result of this assessment, the implementation of the VDA 4965 reference process was further elaborated such that it could encompass other processes than just the ECM.

One key finding when working with the ECM, was that it is important to separate the engineering information artefacts being created and manipulated in the process from the process itself. This means that the process should focus on the actual workflow that it execute and provide message exchange capabilities to and from the workflow to the participants. When looking directly at the VDA 4965 reference process, it turns out that communication, or messages, are exchanged directly between each participant. However, the approach selected for this part of the research, was implementing a collaboration hub in order to provide a level of indirection where each participant only communicated to the collaboration hub. This approach also implied that each participant needed to provide their product data to the collaboration hub in order to enable the other participants to use it. This introduced an unforeseen problem, how much product data is a participant supposed to send to the collaboration hub such that it is useful and provide enough context for the other participants? Or, should a participant just send a minimum of product data to the collaboration hub, but then, the other participant risk to not being able to make well informed decisions since they
can only query a minimum of information in the Shared repository? Moreover, should a participant only interact with the messages being sent and received by the shared processes, or should the participant also separately manipulate the product data in the Shared repository, if so, then what processes should be defined in order to cover a complete collaborative scope? These issues need to be elaborated more thoroughly since they will impact the design of the shared processes. Further, because each participant is passing information from within his own set of enterprise systems to a Shared repository which contains product data from other companies, requires each piece of information to have enough contextual information such that it can be used by the other participants using their own contextual information.

From a software technology perspective, the Process Integration tools play another role than when used in traditional integration projects. Normally the key features of process Integration tools are extraction, mapping and to provide workflow capabilities. In this research standards are used for each message being exchanged to the collaboration hub, thus no need to map the product data in the Process Integration tool since the processes are defined to receive and send messages with payload according to the OMG PLM Services standard, internally the shared processes manipulated the product data in the Shared repository using the PLCS PLM Services. The OMG PLM Services was initially selected for two reasons. Firstly, the Process Integration tool used, Microsoft BizTalk R6, could not be used initially to expose the PLCS PLM Services schemas. Secondly, the VDA 4965 standard is closely related to the OMG PLM Services and the information artefacts defined in the VDA 4965 standards use AP214/OMG PLM Services. However, the conclusion is that OMG PLM Services have to narrow scope to be used for shared processes that should also include other domains and lifecycle phases than just mechanical/manufacturing design. Thus, current research has abandoned the OMG PLM Services. Also for other reasons related to findings when using both the OMG PLM Services and the PLCS PLM Services, there is a need to come up with a new set of implementation standards for web based communication based on ISO 10303 application protocols. Both from a technology perspective as well from a usability perspective, the requirements for this new implementation standard express a need to have messages being sent on the wire that are small and compact. From a technology perspective, the OMG PLM Services and PLCS PLM Services are to fine-grained and introduces network latency, because a user normally needs to make many queries in order to exactly obtain the correct set of information to work with.
Also, the band-width problems appear, since the payload for the fine-grained schemas becomes large. From a usability perspective, messages needs to be defined according to a higher level view of the underlying ISO 10303 application protocol which should also bring the neutral view of the information closer to a user defined contextual view.

5.2 Research validation

The research in this thesis has been performed in order to respond to the thesis statements presented in chapter 1.2.

Standards driven through the Product Data Technology community can and shall be used together with standards originating from the software engineering and technology domain to provide the platform for Model Driven Development in order to build digital information systems that create and manipulate information the way its specified by the industrial domain of concern and to provide implementations that becomes the best of software technology at hand.

In chapter 4.1 and 4.2, Paper A and Paper B evaluated and elaborated with how to setup a three tiered architecture which is based on software engineering best practices. This architecture was then implemented using the ISO STEP implementation methods for data exchange and data sharing. Paper A and B explored and elaborated with various software technology with the goal set to automatically generate all the layers in the architecture, but the findings was that the ISO 10303-214 application protocol only could be used directly as the data layer. The business object layer and the user interface layer needed to process the ISO 10303-214 application protocol in order to compose the fine-grained information entities into larger composite chunks which were then used to define the business object layer and the concepts presented in the user interface layer.

Paper C elaborated further on the three-tiered architecture and managed to generate the data layer based on a modified version of the ISO 10303-214 application protocol. Also, in paper C the data access layer to a large extent defined and generated from the modified version of the ISO 10303-214 application protocol.

At the time of research for Paper D, the Shared repository being elaborated with in Paper C, had extended the underlying information model to also include the ISO 10303-239 application protocol. Also, the findings of Paper D include a draft (submitted to OASIS) of the PLCS PLM Service API that was developed and also implemented on top of the Shared repository presented as
Share-A-space in Paper C. Further, the PLCS PLM Services was implemented as API on other software systems than the Share-A-space in order to evaluate and showcase the use of PLCS PLM Services. To further strengthen the thesis, in Paper E, the use of ISO 10303 application protocols (AP239 and AP214) clearly showed that it is was possible to harmonize and integrate different business processes in a heterogeneous business environment using a Shared repository (based on standards) and Process integration tools that defined their messages using these standards.

Standards are the key enablers for seamless information integration in a heterogeneous system environment where each system is governed by a certain business culture and goal.

The business environment in which the research have taken place and which have been used as this thesis test environment, have always been characterized as highly heterogeneous in terms of different systems used and what business process governs the different systems being used. The only way forward to enable integration of product data between these systems and processes, have been by mapping the information needed to be integrated, using ISO 10303 application protocols. Another approach would have been to define a new common information model that fits the need for all involved systems, but this have never been an option, since it would require large efforts, if possible at all, to re-engineer each involved system and then develop an information model that suited all involved systems. Also, developing a new information model would be a moving target, since systems and processes change, and therefore the model must be updated. Hence, it can be assumed that the supplementary thesis is verified.

Model driven development of digital information systems, based on standards, is a prerequisite for de-coupling information from the digital application systems used to create and manipulate the information.

All software prototypes that have been built and tested throughout the research of this thesis, have all been fully dependent on one, or more, ISO 10303 application protocols. Even if the software prototypes, that implemented the concept of the Shared repository, changed their implementation, the end-user tools that were used to integrate to the Shared repository were never affected. Especially when looking on the commercial Shared repository, Share-A-space, which has been in production since 2001 and have undergone major changes in its implementation, none of the software systems that have been integrated to it
have been affected since the import/export and direct API’s have all been based on the ISO 10303 application protocols and have been kept intact.

Successful application systems for Product Realization implementations requires both use of standards with the focus on software engineering and implementation technology, as well as domain specific standards that focus on information representation.

In the early phases of this research, the implementation methods provided by the ISO 10303 standard were investigated. However, the assessment was that they were too much focusing on file based exchange and was not suitable for when being used to implement for example a data layer that requires high performance. Also, the software technology evolves with a higher pace than the ISO 10303 standards family. For example, when the Web Services technology was accepted and being used by a wide audience, there was (and still is) not enough support in the ISO 10303 implementation standards for how to transform an ISO 10303 application protocol into a Web Service definition. Though, the ISO 10303 community strives to accommodate software standards there will always be a gap, due to the pace. From software engineering perspective, the use of standardized information models have been weak, normally standards for how to represent and implement information models is focused on, for example OMG UML, W3C XML technology, Web Services. Throughout the research, which has been developing prototypes and production software in a business environment, there have always been requirements and constraints that the developed software must fit into commercial software architectures, or parts of it. By stating that this type of business environment is representative for most environments world-wide, would then corroborate the supplementary thesis.

5.3 Conclusions

Based on the previous section, the research thesis is considered to be corroborated. However, the findings found in the research results also illuminates a number of concluding remarks that further needs to be presented.

Indeed the existing ISO 10303 application protocols can be used successfully as a base when developing a complex software system. However, in terms of model-driven development for layered software architecture a number of transformations need to be developed for each particular layer. Also, when implementing a complex software system, the challenging part is mostly how to bind each layer with each other, not just to define the Class and data structures, which is what normally is what is generated in a model-driven development environment.
When using an ISO 10303 application protocol, since it is fine-grained and provides a neutral view in order to encompass for all possible ways to represent product data, there must be a layer above the information model that helps define simplified contextual views, both to serve the users of the information model and the implementers of software.

Reference Data Libraries plays an important role in combination with the ISO 10303 application protocols, but there exist different technologies and different approaches for how to apply them in a software solution environment. The Reference Data Library concept can be used to explicitly represent contextual information such that these can be referenced by the product data that is stored by the ISO 10303 application protocols that provide external data mechanisms.

ISO 10303 application protocols are nowadays accepted by a wider industrial audience, but there is confusion on how to use, and which to use, and how to manage the overlap that exists for the various ISO 10303 application protocols.

Since the ISO 10303 application protocols have started to be adopted by industry as an important enabler for managing the full lifecycle of product data, as well as being positioned internally in an enterprise as the backbone information model, it becomes evident that it would be beneficial for enterprises to be able to provide extensions to the ISO 10303 application protocol. This would make it possible to manage an enterprise information model as a set of small modules that reuse the existing standard.

The existing ISO 10303 application protocols contain a wealth of knowledge for many of the aspects of product data that is still not even envisioned by the commercial PLM vendors on the market. However, due to the nature of the standards, being complex and semantically rich, this makes it difficult for people outside the STEP community to fully understand what the ISO STEP standards really can offer. This also leads to many new efforts in developing new standards, in other communities that many times overlap with what is already available in the ISO STEP standards, which further contribute to the confusion on what standard to use.

As ISO 10303 standards are used not only as an exchange format, but also as something which is used when sharing product data throughout the lifecycle of a product, as well as the lifecycle of collaboration, there are issues to be addressed concerning the information logistics. A company that starts to publish their internal product data externally such that others can reference and update their own product data needs to decide how external change on the
shared information should be managed and how it affects their own business processes.

5.4 Future research

The research started out as a learning experience about ISO STEP and product data applications limited to the design and manufacturing domain. However, early on it was evident that the ISO STEP application protocols provided a much richer representation of product data than what can be found in commercial and in-house product data management systems used today. Therefore it is recommended to pursue the task to find areas of applicability within the product data domain where the ISO 10303 can serve as the core information model. For example, within the area of virtual manufacturing and design it would be possible to include applications that integrated other domains and disciplines that operate on early as well as late lifecycles of the product data.

From a software development perspective the new paradigms of Social computing and Cloud computing impose new possibilities to further increase the use of product data across the lifecycle and enterprise boundaries by new ways interacting with product data and unstructured information.

Since the take up of ISO 10303 standards increase from an industry perspective it is recommended to follow up and elaborate on how to address a unified architecture of the ISO 10303 application protocols. There are initiatives that address this, the STEP modules being used by for example AP239 and AP233 amongst others, is a way forward, but this must be harmonized with the other ISO 10303 application protocols that have not yet adopted to the STEP modules approach.

The use of Reference Data Library technology in combination with ISO 10303 application protocols is also highly relevant. In this area it is recommended to elaborate on how to make it possible to have a unified way to work with Reference Data Library technologies independent on which technology is used to represent the Reference Data. Also, further work needs to be done in order to understand when and where Reference Data is created and how it should be managed, the lifecycle of the Reference Data.

Further, as of today product data is more often created and used by tools and software that is not necessarily traditional COTS PLM, therefore an opportunity would be to exploit how to develop new PLM software that takes full advantage of the product data represented by ISO 10303 application protocols and Reference Data Library technology. This type of new PLM
software could be used to further support the creative engineering process that as of today is not fully supported by PLM software.
6 References


DEVELOPMENT OF INDUSTRIAL INFORMATION SYSTEMS BASED ON STANDARDS


DEVELOPMENT OF INDUSTRIAL INFORMATION SYSTEMS BASED ON STANDARDS


Develop Development of industrial information systems based on standards


DEVELOPMENT OF INDUSTRIAL INFORMATION SYSTEMS BASED ON STANDARDS


