Information modelling for the manufacturing system life cycle

ASTRID VON EULER-CHELPIN

Doctoral Thesis
Stockholm, Sweden 2008
“Essentially, all models are wrong but some are useful.”

George E.P. Box
Abstract

This thesis deals with information modelling within the scope of the manufacturing system life cycle, i.e. the development phase and the operation phase. Information modelling defines and structures information that needs to be managed, and is thereby an important step towards realising efficient information management throughout the manufacturing system life cycle.

The research goal of this work was to find a modelling approach that simplifies management and integration of manufacturing system information, both within and between the development and operation phases. The starting point was an assumption that information integration requires a common information model for the manufacturing system life cycle. The approach was to evaluate the usefulness of the STEP standards AP214 and AP239 (PLCS) regarding how they meet the information requirements. Case studies within the automotive industry were carried out for gathering test data.

Modelling experience showed that PLCS has the most suitable scope since it can represent the manufacturing system from a life cycle perspective. However, the generic character of PLCS introduced other issues, such as how to ensure consistent instantiation. Further guidance is needed regarding how to use PLCS for representing domain-specific objects such as machining centres.

As a response to the inconsistency issue, a concept model of a machining centre was developed to guide the instantiation of PLCS. However, it was found that there are multiple ways to translate the concept model to PLCS depending on viewpoint. Moreover, the characteristics of information management within the operation phase were found to be notably different compared to characteristics of the development phase. For these reasons, it is discussed whether or not a common modelling format for the whole manufacturing system life cycle is appropriate or even realisable.

From a practical viewpoint, it is concluded to be both inevitable and necessary to find appropriate delimitations and interfaces between complementary information models. A promising step towards information integration is to classify the information concepts of different models according to terms defined in concept models.

Keywords: Information modelling, manufacturing resource model, manufacturing system life cycle, STEP, PLCS
Acknowledgments

This work would not have been possible without the contributions and support from numerous of other people. First, I would like to address my gratitude to supervisor Prof. Torsten Kjellberg for his guidance and support throughout this work and for introducing me into the world of Production Engineering. I also wish to address my gratitude towards my co-supervisor for his consultation regarding the PLCS standard, and for taking the time giving me those valuable comments and input necessary for finalising this thesis.

Further, I would like to thank all the co-workers and researchers who have contributed to this work in various ways, by means of collaboration, contributions, interesting discussions, comments, arrangements, knowledge and influences: Andreas Archenti, Jens von Axelson, Mikael Hedling, Dr Sven Hjelm, Patrik Holmström, Prof. Fumihiko Kimura, Mattias Larsson, Christoffer Lindfors, Magnus Lundgren, Dr Thomas Lundholm, Dr Pär Mårtensson, Olof Nyqvist, Dr Johan Richardsson, Dr Daniel T Semere, Dr Gunilla Sivard, Prof. Gunnar Sohlénius, and many more.

My gratitude also goes to representatives of companies and organisations for contributing with time and resources, including: Scania (Södertälje), Eurostep AB (Stockholm), Volvo Car (Torslanda), IVF (Stockholm and Mölndal), Prostep (Darmstadt, Germany), Mazak (Nagoya, Japan), and University of Tokyo – Dept of Precision Machinery Engineering (Tokyo, Japan).

This work has been made possible by the financial aid from Woxéncentrum and the ModArt project within the MERA program. My appreciation also goes to the Swedish Foundation for International Cooperation in Research and Higher Education (STINT), for financing my stay in University of Tokyo.

Finally, I would like to thank my family, Hans, Mineko and Hanna, for supporting me in all thinkable ways, and Joakim for unceasingly standing by me through joy and sorrow.
Definitions and acronyms

Definitions

As-designed, As-planned, As-is p 59
Capability (of a resource) p 48
Concept model p 40
Development phase p 6
Information management p 4
Information model p 4
Information modelling p 4
Manufacturing system life cycle p 6
Model p 26
Operation phase p 6
Runtime data p 46

Acronyms

APxxx Application Protocol xxx (part of STEP) (p 30)
CAD Computer-Aided Design
DEX Data EXchange sets (p 36)
EXPRESS the modelling language defined within STEP – ISO 10303-11 (p 30)
MTBF Mean time between failure (p 49)
MTTR Mean time to repair (p 49)
OEE Overall Equipment Effectiveness (p 49)
OWL Ontology Web Language (p 35)
PLCS Product Life Cycle Support – ISO 10303-239 or STEP AP239
PPR short for “product, process and resource”
RDL Reference Data Library (p 34)
STEP STandard for Exchange of Product model data – ISO 10303 (p 29)
XML eXtensible Markup Language
## Contents

Abstract v  
Acknowledgments vii  
Definitions and acronyms ix  
Contents x  

I  Research Area  

1  Introduction 3  
1.1  Research area .............................................. 3  
1.2  Purpose, goal, research questions, and initial hypothesis ............... 8  
1.3  Outline of this thesis ........................................ 11  

2  Research method and knowledge synthesis 13  
2.1  Development of theories ...................................... 13  
2.2  Information modelling as a research area ........................ 15  
2.3  The research process of this work .............................. 16  

II  Reference Framework 21  

3  Information management within the manufacturing area 23  
3.1  Background: The need for information management ................... 23  
3.2  The role of models ............................................. 25  
3.3  Information models and information standards ........................ 29  
3.4  Knowledge representation and concept models ........................ 38  
3.5  Process/Activity models ........................................ 43  

4  Information management during the operation phase 45  
4.1  Background: The strive for efficiency .......................... 45
### Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2 Runtime data</td>
<td>45</td>
</tr>
<tr>
<td>4.3 Models and standards for runtime data</td>
<td>50</td>
</tr>
<tr>
<td>4.4 Decentralised approaches</td>
<td>50</td>
</tr>
<tr>
<td><strong>III Results</strong></td>
<td>53</td>
</tr>
<tr>
<td>5 Test of standards for representing manufacturing system information</td>
<td>55</td>
</tr>
<tr>
<td>5.1 STEP standards as a starting point</td>
<td>55</td>
</tr>
<tr>
<td>5.2 Test of STEP AP214</td>
<td>55</td>
</tr>
<tr>
<td>5.3 Test of combination of STEP AP214 and STEP AP239 (PLCS)</td>
<td>60</td>
</tr>
<tr>
<td>5.4 Test of STEP AP239 (PLCS)</td>
<td>63</td>
</tr>
<tr>
<td>5.5 Regarding consistent instantiation</td>
<td>69</td>
</tr>
<tr>
<td>5.6 Summary regarding the experience of using STEP</td>
<td>72</td>
</tr>
<tr>
<td>6 A resource model based on AP239 (PLCS) and a concept model</td>
<td>73</td>
</tr>
<tr>
<td>6.1 The need for resource models</td>
<td>73</td>
</tr>
<tr>
<td>6.2 Concept model of a machining centre</td>
<td>75</td>
</tr>
<tr>
<td>6.3 Using PLCS and a concept model for representing a machining centre</td>
<td>78</td>
</tr>
<tr>
<td>6.4 Summary regarding resource models</td>
<td>85</td>
</tr>
<tr>
<td>7 Runtime data and feedback from operation</td>
<td>87</td>
</tr>
<tr>
<td>7.1 Using runtime data</td>
<td>87</td>
</tr>
<tr>
<td>7.2 Runtime data for improving operation</td>
<td>89</td>
</tr>
<tr>
<td>7.3 Runtime data for feedback to development</td>
<td>92</td>
</tr>
<tr>
<td>7.4 Representation of real resources as-is</td>
<td>96</td>
</tr>
<tr>
<td>7.5 Summary regarding runtime data</td>
<td>98</td>
</tr>
<tr>
<td><strong>IV Discussion and Conclusions</strong></td>
<td>101</td>
</tr>
<tr>
<td>8 A modelling discussion</td>
<td>103</td>
</tr>
<tr>
<td>8.1 Structuring a complex multi-dimensional world</td>
<td>103</td>
</tr>
<tr>
<td>8.2 Is one common information model realisable?</td>
<td>106</td>
</tr>
<tr>
<td>8.3 Choosing a model architecture</td>
<td>111</td>
</tr>
<tr>
<td>8.4 Discussion summary</td>
<td>117</td>
</tr>
<tr>
<td>9 Conclusions</td>
<td>119</td>
</tr>
<tr>
<td>9.1 Conclusions regarding research questions</td>
<td>119</td>
</tr>
<tr>
<td>9.2 Conclusions on research goal and initial hypothesis</td>
<td>122</td>
</tr>
<tr>
<td>9.3 Future work</td>
<td>124</td>
</tr>
<tr>
<td><strong>Bibliography</strong></td>
<td>125</td>
</tr>
</tbody>
</table>
Part I

Research Area
Chapter 1

Introduction

This chapter gives a brief introduction to the research area and its focus. The scope, purpose, goal, research questions, and the initial hypothesis are presented.

1.1 Research area

The research area of this thesis involves two domains: the information management area and the manufacturing area (see Figure 1.1). Within this combined area, the overall purpose is to support the activities related to the development and operation of manufacturing systems by the use of efficient information management.

Figure 1.1: The research area: Information management for manufacturing
CHAPTER 1. INTRODUCTION

Information management and information modelling

Information management is related to information science, which has been defined as the science that is concerned with the gathering, manipulation, classification, storage, and retrieval of recorded knowledge (AHD, 2000). Within this work, information management is used instead, defined as follows:

- **Definition:** Information management denotes activities various activities related to defining, creating, structuring, classifying, using, manipulating, storing, organising and communicating information.

Information management as defined above thereby includes the whole process from defining information needs for different activities to the design of computer applications and its manipulation of information. Within the information management area, the main focus of this work is information modelling which is here defined as follows:

- **Definition:** Information modelling is the activity of identifying, relating, and structuring the information types that need to be managed into an information model.

Figure 1.2 shows the scope concerned in relation to other activities included in information management. Areas that are not in focus include implementation-specific models, database and data model design, information manipulation within computer applications, and implementation. The difference between a data model and an information model in Figure 1.2 is that the information model should capture the information needs, whereas the data model is designed from an implementation perspective. Schenck and Wilson (1994) uses a similar differentiation, and adds that even though an information model (unlike a data model) is not explicitly targeted for computer processing, it has the potential for such processing. The following definition of an information model, modified from ((Nielsen, 2003) p.7, p.37) is used:

- **Definition:** An information model is a formal map of required information together with interpretation rules of the used information concepts.

Manufacturing system development and operation – the manufacturing system life cycle

Within the manufacturing area, the main focus is related to the development process of a manufacturing system and the operation of a realised system (see Figure 1.1). The information modelling of this work thereby concerns information related to the manufacturing system life cycle. Before explaining the intended meaning of the manufacturing system life cycle, the life cycles of the product and the resource are first explained.
1.1. RESEARCH AREA

A product’s life cycle can roughly be divided into design, production planning, production, use, and disposal. A machine tool is also a product, but from a user perspective it is a resource that is used to produce other products. Thus, the production phase of the product life cycle and the use phase of the resource life cycle intersect as shown in Figure 1.3 modified from Nielsen (2003).

Before resources such as machine tools can be used in production, the whole manufacturing system needs to be planned. Thus, the scope involves both the production planning phase and the production phase along with the product life cycle in Figure 1.3 (the shaded areas). Those phases constitute what will be called the manufacturing system life cycle, as shown in Figure 1.4. Those two phases will henceforward be labelled the development phase and operation phase of the...
CHAPTER 1. INTRODUCTION

Figure 1.4: The “manufacturing system life cycle” is divided into the development phase and the operation phase

manufacturing system life cycle. The following definitions are used:

- **Definition:** The *manufacturing system life cycle* consists of the development phase and the operation phase, where the *development phase* corresponds to the production planning phase of the product life cycle, and the *operation phase* corresponds to the production phase of the product life cycle which coincides with the use phase of the resource life cycle.

The development phase of the manufacturing system could also be called the design phase of the manufacturing system. However, to avoid confusion with the design phase of single resources, the term *development phase* is used. Information concerning the design and manufacturing of a resource is not included, since the manufacturing resource is regarded from a user perspective, as shown in Figure 1.3.

As shown in Figure 1.4, the development phase is an umbrella term for a set of concurrent and intertwined development processes, such as process planning, resource evaluation and investment, and factory planning. These can in turn consist of several sub processes. In the borderlands of the development phase and the operation phase, there is an installation and a ramp-up phase where the production is tuned in. The operation phase includes, apart from production, activities such as maintenance and improvement work.

The interconnected nature of development and improvement activities, as well as the current need to reconfigure existing systems for new demands, implies that the manufacturing system life cycle does not only consist of one development phase and one operation phase. On the contrary, there are usually several development phases
and operation phases, where a development phase does not necessarily mean the development of a whole factory or line. Instead, other scenarios such as investing in a few resources to eliminate capacity bottlenecks or reconfiguring an existing system for a new product variant are equally common. In short, a development phase can concern both reconfiguration scenarios and investment scenarios for the whole system or only a part of a system. An investment scenario also implies that old resources are disposed of. This is however excluded from the scope of this work.

Problem identification

A manufacturing company’s need to continuously plan for new products and manufacturing systems leads to the need for efficient information management. Manufacturing system development is a complex process where information about e.g. product, process and resources (PPR) needs to be reused for several activities. Although information management is crucial within an organisation, there is also a need to communicate PPR information with external partners, such as resource suppliers and part suppliers. Moreover, there is a responsibility towards customers to provide information about e.g. spare parts and manufacturing information. This is especially important for manufacturers of complex long-lived products, such as automotive vehicles, airplanes and machines.

The operation phase is a phase where a lot of data can be gathered in order to discover deviations or improvement potential. Data can be gathered from various sources during production, e.g. from machine controllers, measurement equipment, and machine operators.

In today’s computerised environment, the actual management of information is often carried out by the use of computer applications where information is created, manipulated, and stored. Along with the increased use of computers, information integration between computer systems has become an issue.

Based on observations from today’s supply of computer applications for manufacturing system development, there is a trend towards more integrated environments where a set of applications are supposed to share data. The overall impression is that there has been much attention regarding information integration within the development environment and the operation environment respectively, but not much between these environments. Thereby there seems to be a gap between the development and operation phases in terms of information integration, as illustrated in Figure 1.5.

This gap has the consequence that there is no updated model that reflects the real resources in the factory. Consequently there is no reliable starting point for manufacturing system reconfiguration activities, e.g. when a new product is to be introduced in an existing factory. In that case, the work of reconfiguring a manufacturing system for new demands does not take the current capability and wear status of resource individuals into consideration. Further, the utilisation of operation experience could be difficult, which means that the ability to learn from earlier mistakes becomes limited.
Figure 1.5: There is not much information integration between operation and development

Vision

What is desired is the ability to access information about the manufacturing system both within the development and operation phases, as well as between them, with a minimum of effort. This would increase the ability to adapt the manufacturing system development to the current situation in the factory, as well as the ability to enable better preservation of knowledge and experience from the operation phase.

Information modelling is needed for defining information needs and can be used for specifying requirements for e.g. computer applications and system architectures. It is thereby an important step within information management. In the end, it is desirable that information is:

- **Accessible**: Information should be easily found, without wondering where to search and how
- **Reusable**: Information should be easily fetched and refined in other applications and contexts
- **Communicable**: Information should be easily exchangeable between various computer applications
- **Interpretable**: The meaning of information should be interpretable for both humans and computers
- **Well-structured**: Information should have its logical place without any ambiguity about where to put it or in what format

Information modelling alone cannot solve all of these aspects, although it plays an important role. For this reason, related information management aspects such as implementation cannot be completely excluded from the discussions.

1.2 Purpose, goal, research questions, and initial hypothesis

The **research purpose** is to aid the specification of data models and system architectures for increasing the accessibility of manufacturing system information within and between the development and operation phases of the manufacturing system.
1.2. PURPOSE, GOAL, RESEARCH QUESTIONS, AND INITIAL HYPOTHESIS

The research goal is to find a modelling approach that would contribute to the accessibility and integration of manufacturing system information within and between the development and operation phases of the manufacturing system life cycle, and come closer to the desired information management characteristics listed in the previous section.

The research questions are formulated using Figure 1.6. There are four things to consider: the models of the manufacturing system (A), the real physical manufacturing system (B), the data that can be acquired during the operation phase of the manufacturing system (here called runtime data) (C), and finally the user who needs and uses information about the manufacturing system (D). The research questions are listed below and numbered according to Figure 1.6.

1. How should resource information be modelled to be utilised for different purposes throughout the manufacturing system life cycle? (A)

2. How can runtime data be used to ensure that there is an accurate model of the manufacturing system? (A + C $\approx$ B)

3. How can experience from the operation phase be utilised in the work processes of all life cycle phases? (A + C available for $\rightarrow$ D)

The first question can be regarded as the overall research question of which the second and third research questions are parts.
Initial hypothesis

An initial hypothesis was stated for this research area:

_in order to integrate information from the development phase and the operation phase, information types from both life cycle phases need to be represented in one information model_

Scope and delimitations

If nothing else is stated, “resource” refers to manufacturing resource with a focus on machine tools, and “process” means manufacturing process – either the type of process (e.g. milling), or a particular process of a factory (process plan), or a particular process operation of a process plan (e.g. a specific milling operation). Further, within this work, “production” and “manufacturing” are used as synonyms.

The main focus of manufacturing system information is information about the manufacturing resources that execute the manufacturing processes, e.g. machine tools. However, since the viewpoint is the user perspective of a resource, the use context needs to be taken into consideration. The use context includes the product that is to be manufactured as well as the manufacturing process. Hence, the term “manufacturing system” emphasises the resource’s close relationship to the process it performs and the product transformation it makes. Other equipment within a manufacturing system, such as conveyors and measuring equipments, are however not considered in this work.

In order to define the research scope further, the following delimitations are made:

- Since the resource is regarded from a user perspective, i.e. for production planning and production (of other products), information particular for the design and manufacturing of the resource is not included.
- The focus is also on information regarding a resource’s involvement in the actual manufacturing process. Thus, information from supporting processes such as logistics, order management, material handling, etc, is not considered.
- The disposal/recycling phase of the life cycle is not included
- This work is mainly about identification of information needs and information modelling. Strategies regarding choice of implementation language, data format, system architecture, graphical user interfaces, etc, are not elaborated.
- Benefits of an information management approach are not analysed from an economical viewpoint.
- The discussion regarding information integration in this thesis is mainly focused around the integration needs within an organisation and its main collaboration partners.
• This work is influenced by factories within the automotive industry in Sweden. In particular, machining systems and machining resources have been studied, but a case from a robot cell of Body-in-White assembly system is also included. The work may however be applicable to other industries as well.

• It is assumed that the problem scenario described in this thesis concerns mainly larger manufacturing companies where information management is crucial.

1.3 Outline of this thesis

In Chapter 2, the methodology is presented, where the research approach is presented and the characteristics of this research area are discussed. The reference framework is presented in Chapter 3-4. Chapter 3 provides a brief background and a state-of-the-art regarding information management. Related work relevant for the operation phase is further presented in Chapter 4.

Results relevant for research question 1 is discussed in Chapter 5 and Chapter 6, and research question 2 and 3 are elaborated in Chapter 7.

In Chapter 8, the results from Chapters 5-7 are discussed and a modelling approach is presented as an idea for the future. Finally conclusions and the need for future work are presented in Chapter 9.
Chapter 2

Research method and knowledge synthesis

In this chapter the approach and research process are described. First, however, a brief presentation of scientific theories is given, and a discussion regarding the characteristics of information modelling as a research area.

2.1 Development of theories

The development of theories has been analysed and described by numerous scientists and philosophers of research methodology. A few of the topics related to this thesis are briefly described below.

Observations

Theories are usually based on facts, which can be gathered by observing the real world. However, Chalmers (1999) states that observations require certain knowledge about what facts to look for, or important facts will easily be overlooked. Thus if observations are to be fruitful, they cannot completely precede the theories that are being formulated.

Inference

Observations play an important role in inference, both in the inductive and deductive approach. In the inductive approach, observations are used to infer a theory. For example, if many observations of heating metal shows that the metals expand, the inferred theory using induction is: “all metals expand when heated”. Deduction is when a prediction or explanation is logically deduced from known theories. Assuming a theory “all metals expand when heated” it can be logically deduced that if X is a metal, then X will also expand when heated. This prediction can be either corroborated or falsified by observing the outcome of an experiment with metal X.
Induction has been target to criticism of not being logically justifiable, e.g. by Popper and Hume (Curd and Cover, 1998). The problem of induction is that the conclusion from the induction is underdetermined by the information in the premises. Because of underdetermination, no conclusion from induction is guaranteed even if all premises of argument are true.

However, as Lipton (1991) put it: “Inductive inference is a matter of weighing evidence and judging likelihood, not of proof”. In this perspective, an inductive reasoning is not objectionable if likelihood is sufficient.

The hypothesis and falsification

According to Popper, there are certain criteria for a scientific status of a theory. One of his conclusions regarding scientific research was: “A theory which is not refutable by any conceivable event is non-scientific. Irrefutability is not a virtue of a theory (as people often think) but a vice” (Popper, 1963). This is the idea of falsificationism, i.e. that a theory or hypothesis must be falsifiable, where a genuine test of a theory is an attempt to falsify it (Popper, 1963).

Falsificationism does not include the problems of induction, since a good theory is not supposed to be proved, only withstand falsification. On the testing of hypotheses, Hempel (1966) wrote: “the confirmation of a hypothesis depends not only on the quantity of the favorable evidence available, but also on its variety: the greater the variety, the stronger the resulting support”. Thus, a good theory is one that has survived many tests of a great variety.

Case studies

Yin (1994) describes a case study as “an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident”. A case study is a suitable method when the context is important, e.g. when analysing why something occurred or how something was developed. A common opinion is that a case study is used when a phenomenon is to be explored. Yin (1994) states however that case studies can be used also to describe or to explain a phenomenon.

Case studies have been criticised for their lack of rigour and the difficulties to make generalisations out of such studies. Some of these problems can be avoided if the case studies are carefully prepared. Further, Yin differs between statistical generalisations and analytic generalisations, saying that the analytical generalisation is possible to make from case studies. Analytic generalisation is to expand and generalise theories as opposed to statistical generalisation where the frequencies within a large set of experiments are calculated.
2.2 Information modelling as a research area

As was described in Chapter 1, the main research focus of this work is information modelling within the manufacturing area. The characteristics of this research area influence both the methodology and the characteristics of the results.

Scientific theories and industrial benefits

From an academic viewpoint, research should aim to establish general theories and knowledge, while research from an industrial viewpoint often aims to increase productivity and profits. Due to the applied characteristics of this research area, the research objectives of this work need to consider both aspects.

The research questions presented in Chapter 1 are closely related to the benefits that are desired from an industrial perspective. The research goal also expresses the need to aid the activities of the manufacturing system life cycle by means of efficient information management.

General theories for information modelling are however not easily formulated in this area, as will be discussed below.

Information modelling and the development of theories

Information modelling typically involves the need to identify what information needs to be handled, and to choose or design an appropriate model for this information.

Having information modelling as a research area makes it hard to apply a deductive approach, since the art of modelling and managing information is not usually expressed by general theories or truths.

An inductive approach is also difficult to apply on information modelling. There is no easy way to synthesise results from different cases and translate them into general rules. One reason is that information models often are tailored to suit specific applications. Attempts to make general statements about information modelling based on specific cases can be problematic, since each case is different. It may be concluded from experience that certain modelling approaches are likely to be better than others, but an overall modelling recipe applicable for all cases is hard to imagine.

Another reason that makes generalisations difficult is that information modelling is a very subjective task. A model depends not only on the purpose of the model but also on the observer/modeller. Further, it is in the nature of information representation that there are no rights or wrongs; there are only imperfect or less imperfect ways, and its level of “goodness” is fuzzy and difficult to evaluate. Thus, there can be several alternative models for the same piece of the world that is described, each with its own strengths and flaws but equally good on the whole.

For these reasons, the purpose of an information modelling approach should not be to formulate a generally valid solution, but to formulate a *good enough*
solution for the concerned scope and the current situation. For example, a certain information model does not claim general validity or rule out other alternative models. Instead, it claims a certain degree of accuracy for the thing it describes, at least for the time being. Since information management needs continuous care and since the real world changes, the suitability of a modelling approach may also change with time.

2.3 The research process of this work

Choosing focus and approach

The initial stage of the research process aimed to create a wide understanding of the domain, as well as to find a research gap. The understanding of a domain influences the path of how to approach the research area. A new researcher usually continues where other people left off, thus avoiding the reinvention of things. However, one should not forget to question what is considered evident or already verified. This work is to a high degree based on results from other researchers, especially regarding the use of STEP standards. The initial hypothesis is partly derived from previous researchers’ results and conclusions.

Information gathering

In the process of understanding the domain, observations from the real world have been of great value. Company visits at manufacturing companies have offered many opportunities to observe contemporary problems in industry. Knowledge of the field has also been acquired from literature, such as books, journals, and papers from international conferences. Participation in international conferences has provided opportunities to get a quick state-of-the-art view of the manufacturing area. Also, the participation in national as well as international projects involving several industrial partners, has also provided insight to industrial needs.

Case studies as information examples

As opposed to the purpose, described by Yin (1994), to use case studies for evaluating something within its context e.g. why and how something occurred, case studies have been used to recreate a situation showing what kind of information was needed to manage during manufacturing system development. In other words, the purpose of the case studies was to gather information in order to recreate development scenarios of the past.

This information was acquired through observations from factories, interviews with employees, and by studying project documentation. Computer applications were also studied to some degree regarding which information was residing in what system, in what format, and for what purpose. The gathered information was then used to test hypotheses regarding the applicability of STEP standards.
However, even if the purpose of case studies was to test a hypothesis, new knowledge and insights can be learned on the way through observations and information gathering. Studies of real cases provide concrete examples of how manufacturing system development and operation can be carried out as well as problems experienced during these activities. New insights are useful for future hypotheses and for the evaluation of information modelling approaches from a practical viewpoint. From this point of view, observations do not exclusively precede theory, and theory does not exclusively precede observations. This goes along with the statement of Chalmers (1999) that in order to successfully observe the real world, certain background knowledge about what to look for is necessary. This background knowledge might as well originate from previous observations.

As a source of information, a case study provides a biased and limited knowledge platform. However, a few case studies can together form a wider knowledge foundation by means of analytical generalisations. The common information types that need to be managed can be identified. As pictured in Figure 2.1, the acquired knowledge can be divided into many abstraction levels. The most specific aspect of case studies concerns the specific information content of each case, e.g. exactly what resources and processes are used and the geometry of the produced product. In the next abstraction level one can discern the general information types that need to be managed and that are common to all manufacturing systems, such as products, processes, and resources. By studying different types of manufacturing systems within the automotive industry, such as machining lines and assembly lines, the common information types become clear. The most abstract level in Figure 2.1 concerns general aspects of information modelling, which are not limited to manufacturing but apply for any area.

It is in the middle level where relevant results can be formulated. The middle level can be seen as the information modelling level, whereas the case-specific level corresponds to the instantiated data. The specific information content in each instance does not provide general conclusions, but is necessary for testing the usefulness of an information model since it defines the requirements of what type of information the information model must be able to handle.

Case studies as a way to falsify or corroborate hypotheses

As mentioned earlier, an important purpose of case studies has been to gather information for testing a hypothesis of whether a certain information standard is good enough or not. Such a hypothesis can either be corroborated or falsified. A falsification is supposed to give an immediate answer, i.e. that the standard is not applicable.

However, the area of information modelling is more fuzzy. A corroborating case, for example, is hard to interpret. There is a difference between “possible” and “good” concerning the use of information standards. When an information standard has been used to represent manufacturing system information, it is seldom difficult to make it possible to represent everything, but it is harder to make sure
everything is represented in a good way without amendments or strained modelling constructions (e.g. by using information types such as “Document” for information that do not fit into the model). When an information model is not good enough, it should be falsified, but the level of what is good enough can be hard to determine.

Regarding the need to consider industrial benefits (see Section 2.2), there is another aspect regarding the usefulness of an information standard. If e.g. an information standard has been corroborated as being sufficient, does it necessarily mean that it is beneficial to use? In other words, does the suggested information modelling approach yield more benefits than the costs of creating and maintaining it? Ideally, the approach could be tested and the improvements, if any, could be measured. However, since the return of an information modelling or management approach may show up first after many years ahead, and since there are probably no exact ways of measuring these effects, this aspect has not been further elaborated. However, the ability to realise and maintain an information modelling approach has been considered to be an important factor throughout this work.

Moreover, since the information amount handled during the manufacturing system life cycle is huge, only small test models have been made. Even if the small models would show potential improvements, it cannot be concluded that the effects are positive in full scale. Further, it does not provide any proof regarding future cases since there are other factors involved.

Another aspect is the need to adapt the use of information models (information standards in particular) to what is being handled. Even if the general information types have been identified, the specific information content shown in Figure 2.1 can bring forth different preferences regarding a model. In other words, even if a certain way of using an information model has worked well in one application, it is not certain that the same approach is convenient for other applications due to the
differences in the applications, or the difference in circumstances.

Test implementations are often an effective way of testing the usefulness of information models, since it adds a larger information management context such as how to store information, what format to use, etc. Unfortunately, the participation in implementation has been limited to one case (the PDTnet case described in Section 3.3).

Research process

The overall research process has been the following:

- Identification of the main information types that needs to be managed within the defined scope.

- Evaluation of the applicability of existing information standards (for the needs identified) using information gathered from case studies. Hypotheses regarding the sufficiency of information standards (STEP AP214 and STEP AP239) have been tested. A fuzzier form of falsificationism has been applied due to the characteristics of this research area described earlier.

- Conclusion of a suitable modelling approach based on the experience from the test of information standards. The reasoning has been based on a mix of facts and modelling experience.
Part II

Reference Framework
Chapter 3

Information management within the manufacturing area

This chapter describes the need for information management and information models, and continues to describe information standards, and other related work of this research area.

3.1 Background: The need for information management

Many manufacturing companies are probably experiencing a trend towards shorter product life cycles and swift market changes. This leads to the need to continuously improve existing products or to develop new product variants. Since it is not economically justifiable to design a new manufacturing system for each new product variant, it has become important to adapt the product design according to what can be manufactured. Moreover, there is a wish to shorten the development time by developing the product and manufacturing systems more concurrently (also known as Concurrent Engineering). Sohlénius (2005), emphasises the importance to treat product design and production design as a single system, with the overall purpose to satisfy customer needs.

A manufacturing company’s need to continuously plan for new products and manufacturing systems leads to the need for efficient information management. System thinking and concurrent engineering put high demands on communication and integration of product, process and resource information. Having the right information at the right time is one of the keys to remain competitive.

Explicit representation of information

Since more and more activities are computerised, computer-interpretable information representations are becoming increasingly important. Explicit representation of information is a necessity for automation and an advantage when information is
to be exchanged between computer systems. It also provides a source for valuable manufacturing knowledge that otherwise could be hidden as human knowledge.

Communication between systems puts higher demands on the interpretation of information compared to human communication. There can be different media of communication, e.g. human speech, physical papers and photos, digital files and images, and structured data files such as XML. From a computer viewpoint, information in structured data files is traditionally easier to process. However, there can be knowledge and experiences that are difficult to represent in a computer-interpretable format. In such cases, speech or paper documents might still be the most efficient way to communicate.

For this reason, the goal of information management should not be to make all information computer-interpretable but focus on information that needs to be shared and reused in various computer applications during the manufacturing system life cycle.

Information management – cost or benefit?

Today’s technology makes it possible to create and store a lot of data at a negligible cost. All stored data is a potential source of knowledge and could be regarded as an asset. However, the organising of data can be troublesome. For this reason, gathered data should have a purpose, although it can be difficult to foresee all possible needs of the future.

It has been stated that information management is important, but one should also remember that information management is an activity that is more or less time-consuming. A carefully designed information system should reduce the time needed to manage information continuously during the engineering processes. If e.g. information models and databases are poorly designed, it would likely lead to poor data quality and time-consuming information management.

Unfortunately, the change of outdated information systems is not easy. Replacements of computer applications and databases are often associated with great costs and the risk of data loss. There is also an insecreteness associated to whether or not the new solution is going to be long-lived or not. The benefits of a new information system could be less than predicted, e.g. due to a complex maintainability, problems to make if fit with the current work processes, or that the circumstances change so swiftly that it quickly becomes obsolete.

For these reasons, the real challenge is to design an information management approach from a long-term viewpoint, that is agile and responsive to changes, easy to use and to maintain. After all, even though information management is a necessity to avoid information chaos, it is also supposed to aid the business – not slow it down.
3.2. THE ROLE OF MODELS

Data and information – sources to knowledge

Information is closely related to knowledge and originates from various data sources. Since the terms “information”, “data” and “knowledge” is commonly used in this field, the use of those concepts within the context of this work will be explained.

Information is often differentiated from data since it has a meaning, e.g.: “Data on its own has no meaning, only when interpreted by some kind of data processing system does it take on meaning and become information” (FOLDOC, 1999). In reality, however, there is no clear distinction between data and information. “Information” in one context might be “data” in the next. For example, news on the radio is information. An attentive listener can acquire knowledge out of it, but for a person who does not understand the language it is uninterpretable data. In other words, even if the interpretation rules exist, the receiver does not necessarily understand them. Davenport and Prusak (1998) then states that the distinction between data and information is decided by the receiver; the intended information is information only when the receiver is truly informed.

However, since there are seldom discrete borders between what is enough interpretation rules, it is not considered important in this work to differentiate information from data. What is more important is the fact that the more interpretation rules and context around a piece of information that is provided, the more knowledge can be acquired from it. Knowledge can also gradually be processed into new knowledge, as stated in (FOLDOC, 1994): “Knowledge differs from data or information in that new knowledge may be created from existing knowledge using logical inference. If information is data plus meaning then knowledge is information plus processing”.

In addition, Figure 3.1 shows how knowledge can evolve further into human competence, where competence is the ability to use the knowledge in a good way (Kjellberg, 1988). Thus, the utmost goal regarding information should, from a human perspective (at least in the context of developing and operating manufacturing systems) be to acquire the knowledge needed to make good decisions. The objective of information management should thereby be to provide an information basis for rational decisions.

Based on this discussion, in this thesis the terms data and information are used arbitrarily in the sense of being the source (or pre-stage) to knowledge, which is the desired outcome of information modelling. In other words, data and information are both sources to knowledge.

3.2 The role of models

For manufacturing companies, the main concern is to work with the development of products, processes and manufacturing systems, and not the development of their information representations. For that reason, it can be questioned how much time should be spent on information modelling. What is the point of creating an information replica of the world? Naturally it would be better to work with reality.
CHAPTER 3. INFORMATION MANAGEMENT WITHIN THE MANUFACTURING AREA

if not the reality had been too complex. To be able to deal with overwhelming complexity, models need to be created where the relevant parts are selected and less relevant parts are cut out. This way, there is a better chance to understand the particular piece of world that is of interest.

Another reason to use models is when the modelled object does not yet exist. Virtual modelling and testing are often less costly than real testing and physical prototypes. There are e.g. many ways to virtually test a machine tool (Altintas, et al, 2005). In fact, the whole manufacturing system development process is about planning, simulation, and testing by means of various models.

The following definition of *model* is used:

- **Definition:** “To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A” (Minsky, 1965)

In the method Structured Analysis and Design Technique, SADT, it is emphasises that every model should have a purpose, viewpoint and detailing level defined (Marca and McGowan, 1988). The model creation entirely depends on what questions the model should be able to answer, who is going to use the model and within which scope. When the model fulfils its purpose within the defined scope, the model has a satisfactory accuracy and can thereby be used as a substitute to the real world for decision-making. The accuracy can however never by 100 % since the only completely accurate representation of an object is the object itself (Davis et al., 1993). Thereby, each model is a deliberate imperfection.
Applying the model definition to the manufacturing area, the real world (A) is the factory to be developed or changed, the models (A*) are the underlying information models used by computer programs, and the observer (B) is the person trying to understand and analyse A. This is illustrated in Figure 3.2, adapted from (Sohlénius, 2005).

**The real world and the mental world**

The distinction between models and the real world is often made, but the mental world is also important. Minsky (1965) states that it is our mental model of the world that we use when we create a model. Thus, a model is a representation of our mental model, transformed into e.g. a drawing, a prototype or an information model. As soon as drawings, prototypes and information models are put into existence, they will be a part of to the real world as well, as shown in Figure 3.3.

This distinction between real and mental is one of the explanations why information modelling can be problematic. If our mental models differ, there can be consensus problems when developing information standards (see Section 3.3), as well as consistency problems when applying such standards (which is shown in Chapters 5-6). The relationship between real objects, mental concepts and models are further illustrated by the meaning triangle described in Section 3.4.

Another issue concerning the mental and real world becomes evident when a model of a real object is embodied into a communicable format, i.e. something more than a thought. Then, the model exists in parallel with the object it describes. If either is changed, the other must be changed as well or it is no longer a good model. Thus, a model of a real object has issues with validity and durability.

Figure 3.4 (Fagerström et al., 2001) can be used to describe this further. Since the model works as a surrogate to the real world when we make decisions, the model
CHAPTER 3. INFORMATION MANAGEMENT WITHIN THE MANUFACTURING AREA

Figure 3.3: Once the model has been created according to our mental models, it is part of the real world as well

Figure 3.4: Models are used for decision-making and must be updated when the real world changes (Fagerström et al., 2001)

needs to be accurate. Therefore, a model describing a dynamic reality requires continuous updating. The decisions will affect the real world, which in turn leads to the need to update the model. If the model is not updated, the model will become obsolete and the decision base will be poor. The more detailed the model is, the more demanding is this problem. Therefore, the detailing level of a model should be in accordance with the model’s purpose and scope, since the efforts of maintaining a too detailed model may exceed the benefits of having such a model.
3.3 Information models and information standards

“Models” within this work mainly refers to information models. An information model has earlier been defined as a map of required information, together with interpretation rules of the used information concepts (see page 4). The interpretation rules means that each information concept as well as relationship has an explicitly defined purpose. An information model also defines how values of attributes should be implemented in a computer system by referring to primitive data types such as String, Real, and Boolean.

Information models can be drawn on a piece of paper, but by representing information models in a computer-processable modelling language such as EXPRESS (STEP Part 11) the model also becomes computer-interpretable (Schenck and Wilson, 1994). When an information model is populated with values on attributes, it is instantiated. More about EXPRESS and STEP is described below.

The STEP standards

During the last decades, computers applications have infiltrated more and more activities which has led to a growing problem of preserving information when software and hardware become outdated. Since each computer application has its own information structure, the export and import of data is often complicated and time-consuming. When information systems are unable to export and import information, they are commonly called “information islands”.

In order to eliminate information islands, there is a vision of a standard information model that provides a neutral format for information. The STEP (STandard for Exchange of Product model data) standards developed within ISO was initiated in the 80s and its purpose was to standardise the information structures of product data to make it independent of computer applications and hardware systems (Al-Timimi and MacKrell, 1996).

A typical scenario where standard models are useful is also when several organisations need to exchange information in a collaboration network. When using a neutral standard format, the number of needed mappings is smaller compared to a scenario where there are point-to-point mappings between each and every application. Figure 3.5 (Ungerer, et al., 2003) shows an ideal situation where a neutral format enables many different applications to share information.

Desirable characteristics of computer systems that the STEP standards aim to address are (Al-Timimi and MacKrell, 1996):

- **Portability**: It must be possible to move information from one system or computer application to another.

- **Interoperability**: In order to facilitate concurrent engineering processes, different applications must be able to share the same set of information and have access to the latest versions.
CHAPTER 3. INFORMATION MANAGEMENT WITHIN THE MANUFACTURING AREA

Figure 3.5: In an ideal situation, systems can exchange data using a standardised information platform (Ungerer, et al., 2003)

- **Longevity**: The information must outlive the application and the computer platform on which it was created. That is important for the ability to reuse and maintain old information that was created in an outdated system.

- **Extensibility**: It is important to keep the possibility of utilising new techniques and continuous evolution.

The STEP standard consists of two main groups: the information models and the tools to create the models (Al-Timimi and MacKrell, 1996). The information models are represented in different Application Protocols (AP) that belong to different application areas. For example, AP214 (ISO 10303-214) defines the “Core Data for Automotive Mechanical Design Processes” and AP239 (ISO 10303-239) is labelled “Product Life Cycle Support”. The tools include the modelling language called EXPRESS and implementation methods such as exchange file formats, e.g. ISO 10303-21 “Part 21” and ISO 10303-28 “Part 28” (based on XML).
3.3. INFORMATION MODELS AND INFORMATION STANDARDS

In this work, the graphical form of the EXPRESS (EXPRESS-G) is used. Simply put, EXPRESS-G models consist of entities (boxes) that represent information types. Entities can have attributes and relationships to other entities. The cardinality of relationships can be defined, telling e.g. that there may be multiple instances but at least one instance (written as [1,?]). Entities can also inherit from other entities and become subtypes. Inheritance relationships are bold. Entities of instantiated models (with populated values on attributes) are often drawn with a line on the lower right corner. The basic principles are shown in 3.6, and the full modelling language is defined in the standard ISO 10303-11 (The EXPRESS language reference manual).

A STEP Application Protocol also consist of an Application Reference Model (ARM) and an Application Implementation model (AIM), which can be mapped together using so called Integrated Resources (IR). In this work, however, only ARM models are used.

STEP AP214 and the PPR information model

Previous research has exemplified that the STEP standard AP214 “Core Data for Automotive Mechanical Design Processes” (ISO 10303-214, 2000) can be used for representing information for manufacturing system development within the automotive industry, such as (Rosén and Johansson, 2000), (Johansson, 2001), (Mårtensson et al., 2002), (Nielsen, 2003), and (von Euler-Chelpin and Kjellberg, 2004). The benefit of being able to represent product, process and resource (PPR) information in a neutral format is emphasised, since this kind of information needs to be communicated between organisations in a manufacturer-supplier network where different computer systems are used. Figure 5.1 in Chapter 5 shows such a PPR structure describing a machining line.
Implementation of AP214 – The PDTnet example

An application based on AP214 has been implemented in the PDTnet project (“Product Data Technology and Communication in an OEM and Supplier Network”) (PDTnet, 2004). The purpose of this project was to facilitate the complicated integration and manipulation of product data by the use of the STEP standard AP214 (Sachers and Ungerer, 2002). It included a number of communication scenarios involving companies within the automotive industry in Germany and Sweden. A special XML schema based on AP214 was developed (the PDTnet XML Schema) for showing information in a neutral web client – the PDTnet Web Client (v2.6).

In the Swedish case scenario within the PDTnet project, AP214 was used to represent PPR information during assembly cell development (Mårtensson et al., 2002). A communication scenario was shown, where information was transferred between systems via a standard-based format. The communication scenario involved a manufacturer and a resource supplier. The scenario is depicted in Figure 3.7: Product data was sent using AP214 in Part 21 format from the manufacturer (to the left) to the resource supplier (to the right). The supplier added process and resource information in their systems and exported the total PPR information according to the PDTnet XML format. STEP files were mapped to the formats of the companies’ systems, which means that import and export functions were needed for the conversion to and from the STEP format. The combined PPR information was then viewed in the PDTnet Web Client (v2.6), as shown in Figure 3.8.

The main contribution to this project was the implementation of process and resource viewing in the web client, which otherwise was used only for viewing product data. As the screen shot shows, there is a process tree on the left side, and the details of the selected process operation is shown on the right side: process type and its description, as well as links to the produced product (input and output) and the assigned resources.

A variety of Application Protocols

AP214 is one of many application protocols in the STEP family. There are several STEP Application Protocols (AP) designed for specific purposes during the manufacturing system life cycle, as shown in Figure 3.9 (Ungerer, et al., 2003). Other examples related to manufacturing system representation include AP239 or PLCS (“Product life cycle support”), AP238/ISO14649 for process control, and AP224 for machining features, and ISO 13399 for cutting tools.

These APs can have partly overlapping scopes, which means that the harmonisation of different APs have become an issue. The most recent application protocols, such as AP239 (Product Life Cycle Support) (ISO 10303-239, 2005) are designed in a modular fashion. The modularisation of the STEP standards aims to increase the harmonisation of different APs by the ability to reuse modules.

---

1 OEM – Original Equipment Manufacturer
3.3. INFORMATION MODELS AND INFORMATION STANDARDS

Figure 3.7: Communication scenario from the PDTnet project using AP214

Figure 3.8: A process tree in the PDTnet Web Client (v2.6 from 2004) – an early viewer for AP214
CHAPTER 3. INFORMATION MANAGEMENT WITHIN THE MANUFACTURING AREA

Reference Data Libraries (RDL)

A STEP AP cannot be too specific since it is supposed to be applicable for all cases within its purpose and scope. Too specific information types would likely make it unsuitable for some applications. Moreover, it would become sensitive to changed circumstances, such as the introduction of new resource types and tool types.

The generic character of a STEP standard entails that each organisation can choose to instantiate information in a slightly different way. Both (Johansson, 2001) and Nielsen (2003) concludes the importance of a consistent way of naming and classifying entities when using the standard AP214. Since it is very unlikely that each modeller chooses an identical modelling approach, the communication between organisations, or even within organisations, cannot be fully automatic without agreements on the instantiation.

In order to make information standards more specific without losing their generic qualities, the newer APs such as AP239 (PLCS – Product Life Cycle Support) and ISO 13399 (Cutting tool data representation and exchange) use further definitions of concepts outside of the core model in so called reference data libraries (RDL) (Larsson and Kjellberg, 2006), (Nyqvist and Kjellberg, 2004). Reference data classes are referenced from the core model to classify entities into more specific information types.

Reference data can have different levels of complexity (Eurostep, 2005), (Larsson, 2006). The simplest form could be a list of definitions. A complex form of reference data is to define an ontology model that specialises the generic model. The standard ISO 15926 (Integration of life-cycle data for process plants including oil and gas production facilities) (ISO/FDIS 15296-2, 2003) includes an initial set
The RDL defined for PLCS lies in the middle. It allows further specialisation of generic information into more specific reference data classes, but do not define relationships between them. The standard RDL for PLCS is represented in OWL format. OWL (Ontology Web Language) is an XML-based format for publishing and sharing ontologies on the web (W3C, 2004).

The use of RDL together with the core model PLCS is exemplified in Figure 3.10 (a part of Capability C001 (OASIS PLCS TC C001, 2007)): There is a Part with an identifier “MACHINE 1” which is used (or owned) by an Organization with an identifier “ORG 1”. The Identification_assignment instances are further specialized into the reference data classes Part_name and Organization_name. Organization_or_person_in_organization_assignment is classified as the reference data class called Owner_of.

Data exchange sets (DEX) and Capabilities – communication using AP239

In order to use generic STEP standards such as PLCS for system-to-system communications, further guidance is needed in addition to the core model and reference data libraries.
In order to know what modules of the PLCS standard should be used, so called Capabilities are defined to describe typical instantiations, e.g. “representing product as individual”. Capabilities are used to ensure a common interpretation of PLCS, to avoid multiple dialects of PLCS, and to simplify instantiation of the PLCS data model (OASIS PLCS TC, 2007). A part of a Capability (C001) is shown in Figure 3.10. Within a capability, so called Templates are defined, specifying exactly what entities need to be represented.

Based on these Capabilities, Data Exchange Sets (DEX) are defined. A DEX specifies a set of information that is to be exchanged to support a particular business process, e.g. managing “product as individual” (OASIS PLCS TC, 2007). Entities in Capabilities and DEXs can refer to reference data for defining and classifying the entities. The work of defining Capabilities, DEXs and RDL is a standardisation work by the PLCS Technical Committee within OASIS (Organisation for the Advancement of Structured Information Standards) (OASIS, 2008). Ever since the core model of PLCS reached the international standard status in 2004, the work of developing DEXs and RDLs has continued within OASIS.

The communication scenario can be pictured as in Figure 3.11, adapted from (Eurostep, 2005). In order to set up a communication scenario based on PLCS and DEX, there also need to be project-specific business agreements on the Capabilities, reference data, as well as exchange agreements regarding DEXs between the communicating organisations (Eurostep, 2005). The exchange agreements concern what parts of a DEX is needed, which RDL to use, adaptations of Capabilities to what is communicated (business concepts), and the file format to be used for the exchange (e.g. Part 21 or Part 28).

Since the setup of this type of communication scenario requires a relatively high initial effort, this kind of communication will likely be used only for long-term exchange scenarios with a high degree of reusability. The work within OASIS is however still under progress.

Other standards
Besides the STEP standards, there are other information standards of various abstraction levels and scope related to manufacturing.

The information types of ISO 15926 (Integration of life-cycle data for process plants including oil and gas facilities) starts from a philosophical level with a universal supertype called Thing – the class of everything (ISO/FDIS 15296-2, 2003). It decomposes into possible individuals and abstract objects, each of which decomposes further into various phenomena or occurrences that exist in time and space, such as fluids, gases, etc. Due to the generic character of the model, a reference data model has been defined within this standard, providing more useful entities for representing process plants.

Other standards have a wider scope, such as models and frameworks on the enterprise level. CIMOSA, for example, defines a three-dimensional modelling framework (Kosanke, 1995). One axis shows different views: the organisation, resource,
3.3. INFORMATION MODELS AND INFORMATION STANDARDS

information and function views. Another axis shows the definition stage: requirement definition, design specification and implementation description. The third axis is for the instantiation levels: generic level, partial level, and particular level. In other words, a manufacturing system model is also a part of a larger enterprise context.

More specific standards defining machine tool information are under development, both within STEP-NC Part 110, and within ASME (The American Society of Mechanical Engineers). An initial machine tool model related to STEP-NC Part 110 has been described by Vichare et al. (2007).

The upcoming standard ASME B5.59-2 (Data specification for properties of machine tools for milling and turning) is an XML-based information model and data format for properties of machine tools for milling and turning (ASME B5/TC56, 2005). ASME B5.59-2 addresses the performance and capabilities of the machine tool at any time in its life cycle, e.g. during specification, after acceptance testing, or at any time during operation (ASME B5/TC56, 2005). It uses specific information concepts used for specifying and analysing machine tools, such as work zone, axes, spindles, table, and controller, installation, performance, machine errors, etc. The context of the information in a B5.59-2 data file depends on the value of the tag <INFORMATION_STATUS>, which can have values such as: specification by customer, specification by vendor, acceptance test from vendor, acceptance test from user, immediately after collision, immediately after moving, immediately after rebuild, or after repair. Thus, the differentiation of life cycle stage and context is addressed by a string value of a single tag that tells how to interpret the rest of
the information content. ASME B5.59-2 only concerns information related to the machine tool itself and does not include process-related information.

Pros and cons of information standards

Standardisation has the positive effect of forcing explicit definitions of information types and thereby contributing to a common frame of reference. Standardisation of information models is often a long and time-consuming process and a compromise of many opinions. Even if there are different opinions of how information should be structured, a finished standard represents a common base of consensus which is useful to refer to.

The use of standards as an intermediate communication format also reduces the number of translations between applications compared to a point-to-point solution.

Standards are also subject to criticism, such as being designed according to yesterday’s information needs. A finished standard is not future-proof and needs to be continuously revised. Moreover, by the time one standard is finished, there may be “better” standards on the way. Generic information standards are however likely to last longer than specific information standards.

Another issue is the wide range of standards with overlapping scopes, but with different detailing levels or viewpoints. Since each standard has its particular purpose, viewpoint and detailing level, the use of several standards may result in “standard islands” which perhaps is not much better than a situation where standards are not used at all.

Furthermore, standards are often too generic to be able to catch the need of specific organisations. The need for further guidance of how to apply standards can give the feeling of unnecessary complexity, especially if it is used for an isolated case.

As with most things, however, the use of information standards can either be fruitful or costly depending on how standards are applied. It depends e.g. on which standard is used, what parts of the standard are used, and how it is applied. For example, the use of standards is likely to be beneficial when many different information sources need to communicate during a long period of time.

3.4 Knowledge representation and concept models

Knowledge representation

In some applications, knowledge representations are needed rather than information models. Whereas information models are focused on how to structure information entities, knowledge representations rather aim to represent known facts of the real world. The represented knowledge can then be used for logical reasoning.

When representing knowledge of a domain, the term ontology is often used. Ontology has been defined as the science of the concepts or categories needed for making a coherent, consistent and exhaustive description of a part of the real world.
Ontology has also been described as the study of existence (Sowa, 2000), i.e. it describes what exists in the world. While philosophical ontologies often start with an universal supertype, such as “Thing”, ontologies used for computing usually start from bottom up (Sowa, 2000).

There are examples of ontologies within the manufacturing area, e.g. PSL for describing manufacturing processes (Pouchard et al., 2000), and ontologies for describing resources, skills, and processes (Onori et al., 2006), (Semere et al., 2007). Such ontologies makes it possible to reuse knowledge about available processes, resources, features, etc for knowledge-intense processes such as process planning.

Ontologies often include classification hierarchies. A class is an abstraction defined by its members (NE, 2006). Within the manufacturing area, a process taxonomy is presented by (Todd et al., 1994). A resource directory of available machine tools is another example (Rote Buch, 2008).

A tree-shaped classification does not allow multiple inheritances. However, things in the real need often be classified in terms of several aspects. For example, a yellow robot could belong to the category of robots in a resource classification and the category of yellow objects in a separate colour classification.

A knowledge representation is, as with all models, a surrogate for the real world and thereby a simplification where less important parts are cut out. Davis et al. (1993) also brings forward other effects of choosing a certain knowledge representation. For example, the choice of representation technique (logical rules, frames, semantic nets, etc) affects what aspects of the real world that are represented, since each representation technique has characteristics regarding what and how to model the world. The representation technique also affects how reasoning is carried out. In order to make use of a representation technique, it should be used as it was intended to be used (Davis et al., 1993).

**Concept models**

An information model is usually created in order to structure information and define information types. When representing knowledge, however, the focus is rather on the meaning of concepts rather than the structuring of information.

A way to describe the meaning of a concept is to relate it to other concepts. Such models will be referred to as concept models. In this work, concept models are drawn based on the ASTRAKAN modelling format (ASU, 2003). As Figure 3.12 shows, concepts – symbolised by terms – are connected via relations. When relations are of a specialising character, a circle is drawn connecting the specialised concepts together.

The concept models used in this work can be compared to the conceptual graphs defined by Sowa (2000). Sowa’s conceptual graph is a bipartite graph that has two kinds of nodes called concepts and conceptual relations. The concept models drawn in this thesis are however simplified compared to conceptual graphs, since the terms of the relationships are textual and not drawn as separate concepts. Moreover, the concept models presented here are not (yet) intended for computer reasoning.
the contrary, a less formalised format is used to be able to focus the meaning of concepts and find appropriate terms symbolising them.

Within the context of this work, a concept model is defined as follows:

- **Definition:** A concept model is a graph for the purpose of explaining a concept (symbolised by a term) by its relationships to other concepts (symbolised by other terms).

Concept, terms, and objects

A concept refers to an idea or thought. When humans communicate a concept, symbols and referents are used. Symbols are e.g. terms/words and referents are examples of the world. Ogden and Richards (1936) describes these three factors involved in human communication as a triangle in Figure 3.13. It has later been described as the meaning triangle. These factors are related since the choice of symbols to express a thought is partly caused by the objects we refer to.

Thought can also be described as concept and referents as objects (Sowa, 2000). Objects can be seen as instances of a concept. A concept instance is however not the same as information model instances, since concept instances are instances of the real world, while information model instances are only instances within the scope of an information model.

Levels of knowledge – the Zachman framework

According to Sowa (2000), levelling of symbols was illustrated by Charles Sanders Pierce by linking meaning triangles together. When a symbol is an object of another symbol, a horizontal linking is formed. As an example, Sowa showed the resemblance of horizontal linking with the different design levels of the Information System Architecture (ISA) presented by Zachman (1987). A modified example from Sowa (2000) is shown in Figure 3.14.

The Zachman framework shows different types of descriptions of a product depending on the perspective. The question what leads to e.g. data descriptions, how to process descriptions, where to business networks, who to organisation charts,
3.4. KNOWLEDGE REPRESENTATION AND CONCEPT MODELS

Figure 3.13: The triangle of thoughts, symbols and referents, adapted from (Ogden and Richards, 1936)

Figure 3.14: Pierce’s linked meaning triangles applied on a part of Zachman’s design framework, adapted from Sowa (2000)
when to schedules, and why to strategy. These distinctions can be applied to various hierarchical levels of an enterprise. The result is a matrix, where the columns represent the abstractions inferred by the six questions listed above, and the rows are perspectives of an enterprise: planner, owner, designer, builder, and sub-contractor. The planner’s level includes strategy and the owner’s level includes business models. Below these levels, it starts to involve information structures and implementation. The system level corresponds to e.g. information models, the technology level takes care of implementation aspects (i.e. data models), and at the bottom level – the component level – the actual programming specifications can be found.

As can be seen, models can be created from many perspectives in various levels of abstraction. The work of this thesis can be associated to the question what at the system level.

A similar framework adapted to the information management area has been presented by Larsson (2006).

Implementations of concept models

Regarding the actual implementation of concept models, there are several variants. Traditional formats are e.g. dictionaries and encyclopedias providing a textual explanation of each term. By the use of web-based technology, such encyclopedias can be made navigable (see e.g. Wikipedia\(^2\)).

An alternative that is less textual and more graphical is the navigable concept browser Conzilla\(^3\) presented by Naeve (2001). In this concept browser, a concept has both content and context, where the content is the textual definition or description of the concept, and the context consist of related concepts. Each concept within a context map can have its own content as well as context map. Thus, the result is a navigable context map. One could compare it to moving the focus of a camera; when one object is in focus, the surrounding objects become blurred (contextual). Relations between concepts can be of different types, such as “kind of”, “part of”, “is a”, and “aggregation”.

Knowledge in systems

Computer applications are generally based on an information model that structures the information types handled by the application. For the purpose of good data quality or for avoiding computation errors, basic knowledge is usually implemented in order to warn when improper formats or values are used. However, in most computer applications used for manufacturing system development, the degree of built-in engineering knowledge and manufacturing experience is limited. The system does not evaluate the choices of the user, and thereby it does not know if e.g. a process planner is skilled or inexperienced. Unlike expert systems which are

\(^2\)see http://www.wikipedia.org/
\(^3\)see also http://www.conzilla.org
designed to aid in thinking and reasoning, “ordinary” systems handle information rather than knowledge.

Nevertheless, there is often, perhaps always, some degree of built-in knowledge inside an application. The information model itself expresses to some extent knowledge about the represented concepts or objects by their structuring. Moreover, a process planning tool could have a library of available processes, tools, and manufacturing features. Other examples of built-in knowledge is formulas or algorithms for generating tool motions or for calculating cycle times of process operations.

Since knowledge often is integrated with the application logic, it is seldom reusable to other systems. However, resource, process, and feature libraries have a high potential of reusability if they were defined independently. An external manufacturing ontology including libraries of available resources, processes, features, etc could be used by many systems. However, such sharing of libraries among systems seems to be limited.

3.5 Process/Activity models

As was defined in Section 1.1, information modelling is focused on what information needs to be managed. Implementation aspects further define how this information is to be managed. Other questions recognisable from the Zachman framework are also when and why, i.e. when information is used for what purpose. These perspectives can partly be captured by defining process models. Such models can define in what activities information is used, manipulated and created. There are a number of representations available for representing processes and activities, e.g. within the IDEF (Integrated definition methods) and UML (Unified Modelling Language) families.

A process in this context does of course not refer to a manufacturing process. A process in this context has been defined as a pattern within which value-adding activities are performed (Nilsson, 2004). Value-adding means that the output of an activity has a refined state compared to the state it had as an input.

In the ongoing project ModArt (ModArt, 2007), process models are being defined for four main sub processes within the manufacturing system development and operation phases: process planning, factory planning, resource investment, and improvement work. These processes are not sequential and independent but rather parallel and intertwined. The modelling format used for these processes is based on a format within the ASTRAKAN group, described by (Nilsson, 2004). It follows the same principle as IDEF-0 (Integration definition for function modeling) (NIST, 1993) in that activities can have input, control, output, and supporting mechanisms, and that activities can be decomposed into other activities.

Within the ModArt project, the process models have been implemented as web-based process pilots. Figure 3.15 shows an activity (green arrow) translated as “Selection of machine individual”. The output of this activity (the yellow boxes to the right) can be of two types. If there is an existing factory, the output of this
activity is “selected resource individual”. An alternative output is “needs for new resource”.

Process models describes one aspect of information that cannot be described by information models alone, i.e. how information evolves as Lutters (2001) expressed it. Process models catches the time difference between the modelled information, indicating e.g. that there must be a product model before a process model can be created.

Even though the focus of this work is the modelled information, it is necessary to know who is going to use the information for what purpose in order to understand what, how and why information needs to be modelled.
Chapter 4

Information management during the operation phase

This chapter gives a reference framework regarding information management during the operation phase of the manufacturing system life cycle.

4.1 Background: The strive for efficiency

When manufacturing resources have been installed and configured, the focus is to run the system as efficiently as possible. Many manufacturing companies aim to increase efficiency by reducing unnecessary cost according to the principles of Lean Manufacturing and the elimination of waste, described by e.g. Ohno (1988). The Japanese manufacturing company Toyota developed their “Toyota Production System” into a world-known model of Lean Manufacturing that many manufacturing companies aim to live up to. Lean Manufacturing is characterised by minimal buffers and Just-In-Time deliveries.

A slimmed system is a key to become profitable, but it also relies on a deep commitment and carefulness among all workers. Any disruptions in the information flow will lead to swift disturbance chains (Ohno, 1988). Moreover, disruptions are supposed to be discovered and not hidden. Thus, the more information that is available about the status of the resources and the production, the higher is the possibility to act pro-actively and improve the overall performance.

4.2 Runtime data

The strive for maintaining and improving efficiency means that data from the production needs to be collected, either manually by operators or automatically using monitoring systems, sensors, machine controllers, etc. All information that can be gathered from the runtime environment related to the actual manufacturing will
henceforward be included in the term runtime data. The following definition is made:

- **Definition:** Runtime data include all types of data that can be gathered during operation, as well as information synthesised from this data, that are of relevance for evaluating the capability of a real resource.

**Life cycle data**

During the last years, there has been an increased focus on the product life cycle. It is especially an issue for complex products with long life cycles such as airplanes and trucks. A manufacturing resource is also a complex product whose life cycle should last as long as possible.

The current trend within manufacturing, at least in the automotive industry, is that the life cycle of manufacturing resources outlives the life cycle of the products they produce. The reconfigurability of manufacturing systems is thereby becoming increasingly important. Flexible manufacturing resources such as robots and machining centres can do various types of processes depending on how they are configured, which means that they are more easily reused for new processes. Therefore it is important to take care of life cycle data about such resources, since wear and status information can be important during reconfiguration.

Within this context, life cycle data means information that tells how a physical product changes from its original state during its use phase. For a manufacturing resource, life cycle data include e.g. replaced parts, maintenance, and wear indications such as capability and accuracy information.

Another reason to collect life cycle data and monitor the resources is to prevent machine failures. According to the principles of TPM – Total Productive Maintenance (Nakajima, 1988), preventive maintenance is important in order to avoid unplanned maintenance and machine failures which usually are more costly than the preventive maintenance.

**Resource capability and behaviour**

When the word capability is used within manufacturing, it usually refers to the process capability index ($C_{pk}$) which is a statistical measure describing the outcome of a process compared to a given tolerance interval (Juran and Gryna, 1988). Juran defines process capability as the measured inherent reproducibility of the product turned out by a process. This capability is measured in terms of $6\sigma$, where $\sigma$ is the standard deviation of the process under a state of statistical control, i.e. with no major disturbances or major changes.

Another variant is machine capability which tells the reproducibility under a set of process conditions that are as stable as possible. Process capability, on the other hand, is measured during a longer period of time and takes the normal changes
4.2. RUNTIME DATA

Figure 4.1: EXPRESS model illustrating a resource’s general capability to accomplish manufacturing features

in process conditions, workers and material into consideration (Juran and Gryna, 1988).

Within this work, however, capability is not regarded from a mathematical viewpoint but from a modelling viewpoint. In other words, the concern is to be able to model all information needed for describing several aspects of a resource’s capability.

From the modelling viewpoint, the term capability needs to denote a wider concept than reproducibility. A resource can have certain capability when used in a process, but it has also some basic capabilities as a resource. The basic capability of a resource, e.g. a machine tool, is expressed by the types of manufacturing processes it supports, and thereby which product transformation it is able to perform on a material (Holmström and Kjellberg, 2004). For instance, if the machine tool is a drilling machine, it can do round holes in a material, since the drilling process is associated with the round hole feature. Figure 4.1 shows a model where a resource is capable of creating manufacturing features through a manufacturing process, modified from (Holmström and von Euler-Chelpin, 2004).

A machine tool’s capability is further defined by the properties of the machine tool (e.g. number of axes) and its components (e.g. spindle speed, feed force). Depending on such properties, the machine tool is able to machine various features of various sizes, with different accuracy and speed.

This general aspect of capability is useful in the situation where manufacturing systems are to be reused and adapted to several product variants with variable production volumes. By analysing the allowed processes (and resulting features) as well as properties of the resources, the choice of product variants can be adapted to minimise reconfiguration of the system (Kimura and Kakuta, 2005), (Kimura and Nielsen, 2005).

Whereas the basic capability of a resource in isolation can be described by en-
abled processes and properties, other aspects become important when the resource is put into a specific process. A machine tool’s expected capability in a certain process depends on the current process conditions, e.g. used tool, material, the size of the feature, temperature, etc (Holmström and Kjellberg, 2004). Some external influencing factors relevant when evaluating capability have been identified in the upcoming standard ISO/CD 26303 (2006):

- Process information such as feature/tolerance, process parameters, cutting forces, tool accuracy, tool wear, lubrication, etc
- The machine tool’s geometric, static, dynamic and thermal characteristics
- The characteristics of the material of the machined part
- Factors concerning operators and shifts
- Environmental aspects such as temperature changes, vibrations from surroundings, etc

Conclusions regarding a machine tool’s dynamic behaviour can also be drawn by analysing the produced output. Product accuracy can be used to analyse the geometrical accuracy of the machine tool, such as angles between axes, spindle direction in relation to the axes, parallelism, and positioning accuracy. Tests of dynamic stiffness such as described in (Archenti and Nicolescu, 2007) show that a machine tool’s behaviour is not only based on its parts on properties, but also by its runtime behaviour. In other words, from some perspectives, the appropriate representation of a machine tool’s capability could be in the form of diagrams and graphs.

In order to clarify what aspects are included in the term *capability* in this work, the following definition is made:

- **Definition:** A resource’s *capability* is about having the right components, properties, and behaviour to be able to create a feature (or other value-adding transformations on the product), with a certain quality and within a certain time.

There are three aspects of capability, depending on the life cycle stage:

- The *general* process-independent capability of a resource can be derived from the physical and functional properties of a resource. Examples are properties and components found in resource specifications, such as spindle speed, work zone, and number of axes.
- The *expected* process-dependent capability of a resource refers to its estimated ability of performing assigned operations based on the current configuration and tolerance requirements. Examples are expected cycle time and expected product accuracy.
4.2. RUNTIME DATA

- The *observed* capability of a resource can be derived from the behaviour and statistical measures calculated during runtime. Examples are observed cycle time, and various performance measures.

Capability information can be used for various purpose throughout the manufacturing system life cycle. During manufacturing system design, the decisions regarding the reuse of resources can be based on the measured accuracy of resource individuals. During product design, the manufacturability of future products can be evaluated based on the measured accuracy of current products produced by the existing resources. During resource design, it can be imagined that models presented by Altintas, et al (2005) for virtual analysis of a not yet realised machine tool can be enriched with runtime data from a real machine tool.

Performance measures

As a part of the wide definition of capability just described, performance measures are one aspect, i.e. how the resources perform in relation to expectations on product accuracy, cycle times, availability, etc. There is a wide range of performance measures, but all of them are not elaborated here. Instead, a few typical performance measures will be used as examples. A widespread measure is Overall Equipment Effectiveness (OEE), which is a simple equation taking both time aspects and quality aspects into focus. OEE is defined as (Nakajima, 1988):

$$OEE (\%) = \text{Availability} (\%) \times \text{performance efficiency} (\%) \times \text{rate of quality products} (\%)$$

*Availability* is a measure that tells how much of the time the resource has been running compared to the time the resource was scheduled to run. Many disruptions will bring this rate down. *Performance efficiency* compares the ideal time to produce a number of items in relation to the actual time it took. The *rate of quality products* tells the percentage of approved items out of the total processed amount. OEE can be calculated for a system as well as for a single resource.

The availability of a resource can also be calculated as:

$$\text{Availability} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

where MTBF is the “mean time before failure” and MTTR “mean time to repair” (when parts are available) (Nakajima, 1988).

Sometimes performance measures are expressed as requirements or goals. There could be general goals such as maximum errors per produced product, throughput time of a product, etc. There might be goals to attain lean manufacturing and reduce waste, and goals such as fulfilling satisfactory levels of ergonomics/safety, environmental care, and flexibility. These types of goals are often put on the total system and are hard to translate into goals for each operation or on a single resource. In this work, only performance measures that can be associated to a
resource’s capability to produce a product are considered. Typical examples are those measures needed for flow simulations and capability evaluations, e.g. MTBF and repeatability. Whatever the performance measures are, the objective is that necessary data are captured during runtime, so that various aspects of resource capability can be analysed.

4.3 Models and standards for runtime data

There are many examples of standards that address a resource’s behaviour during operation. The upcoming standard ASME B5.59-2, previously described in Section 3.3, also includes performance and accuracy aspects of machine tools for milling and turning (ASME B5/Tc56, 2005). Another upcoming standard is ISO 18435 (Diagnostics, capability assessment, and maintenance applications integration) which covers the use of operation data to improve execution and manufacturing system design (ISO/WD 18435-3, 2004), (ISO/CD 18435-1, 2006). Another initiative is ISO 15531 (also called MANDATE) which also defines information regarding the usage of production resources (ISO/CD 15531-32, 2001). An recent initiative within STEP-NC (STEP AP238/ISO 14649) defines machine tool data for the purpose of traceability, i.e. a model used for representing motions before a failure, comparable to information monitored in the black boxes of airplanes.

Standards related to flow simulations include NISTIR 7198 – Machine Shop Information model (NIST, 2005) and Core Manufacturing Simulation Data Information model -(CMSDIM), (SISO, 2006).

As can be seen, there is a variety of standards covering various aspects of the operation phase of the manufacturing system life cycle. Some initiatives are narrow and detailed within the operation phase of a manufacturing system, whereas other cover aspects from both the operation phase and the development phase of the manufacturing system, e.g. ASME B5.59-2 and STEP AP239 (Product Life Cycle Support).

4.4 Decentralised approaches

For many years, “lean” has been an important keyword for how to run a manufacturing system. However, since life cycles of products is shortening, it has become important to be able to reuse manufacturing systems for new products. Consequently, the strive has become to make the systems “flexible”, “reconfigurable”, and “agile” to changes. One way to bring flexibility into systems is to use flexible resources such as machining centres. Another way is to exploit modularity into evolvability.
4.4. DECENTRALISED APPROACHES

Evolvable manufacturing systems

In the field of micro assembly, a new concept called “evolvable” manufacturing systems has emerged in order to cope with the fluctuating market needs (Onori et al., 2006). Evolvable manufacturing systems exploit the phenomenon of “emergent behaviour” and use influences from biological systems. For example swarms can show emergent behaviour when they accomplish greater unforeseen effects through interaction and cooperation. Emergence is not automatically achieved by assemblies – it is the interaction between the parts that provides the explanation (Minsky, 1988). This indicates that a set of simple objects with modest intelligence can create new behaviour, if they could interact, i.e. receive signals, and adapt to the new circumstances.

To make a manufacturing system evolvable, it means that the approach of investing in advanced, flexible units that can be reconfigured to do almost everything is replaced by an approach of having smaller and less advanced modules with simple computational abilities that can interact with other modules and tell about their skills/capability and status information. In that way the modules can self-organise, cooperate and create new behaviour. This idea requires a manufacturing ontology which includes a categorisation of equipments and a description of their skills (Lindberg, et al., 2007), (Onori et al., 2006).

BMS – Biological Manufacturing System

Another manufacturing system concept representing similar principles is Biological Manufacturing Systems (BMS). The idea of biological influences in computing and information technology has been present during a long time, in research areas such as Artificial Life simulations, genetic algorithms, and agent programming. This has also influenced the manufacturing area. BMS addresses biologically inspired ideas such as self-recognition, self-organisation, adaptation and evolution (Ueda, 2006).

The need for BMS thinking originates from the increasing complexity and uncertainty arising from factors such as globalisation of industries. The situation is characterised by incomplete data and knowledge, a vast number of possibilities and combinations, dynamic changes in environment, etc (Ueda, 2006).

Also BMS uses an emergent approach rather than a deterministic one. The environment that affects the manufacturing system is difficult to predict and simulate in a trustworthy way. Thus, instead of trying to predict the market and control the system from a top-down perspective, the emergent approach suggests that the manufacturing units should be able to adapt to new circumstances. The differences between the deterministic and emergent approaches are described in Figure 4.2 (Ueda, 2007).
### Emergence and information management

Traditional information management assumes that all information types are known beforehand, and the information model is designed to cover the information needs from the beginning. However, if the future manufacturing system is evolvable, the representation of the real system must also change accordingly, in a decentralised fashion.

A decentralised approach would however not eliminate the need for traditional information modelling since it is unlikely that everything can be solved by the evolvable principle. Moreover, even if models of small modules are simpler, there is also a need for a common ontology and a common communication protocol for the communication to work. Thus, it is likely that simplifying effect of one aspect (e.g. the size of each information model) is compensated by a complicating effect of another (e.g. establishing a common ontology).

<table>
<thead>
<tr>
<th>Deterministic approach</th>
<th>Emergent approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top down</td>
<td>Bottom up</td>
</tr>
<tr>
<td>Whole to parts</td>
<td>Parts to whole</td>
</tr>
<tr>
<td>Outside to inside</td>
<td>Inside to outside</td>
</tr>
<tr>
<td>Closed</td>
<td>Opened</td>
</tr>
<tr>
<td>Concentrated</td>
<td>Distributed</td>
</tr>
<tr>
<td>Global determination</td>
<td>Local determination</td>
</tr>
<tr>
<td>Direct control</td>
<td>Indirect control</td>
</tr>
<tr>
<td>Serial processing</td>
<td>Parallel processing</td>
</tr>
<tr>
<td>Explicit process</td>
<td>Implicit process</td>
</tr>
<tr>
<td>Static environment</td>
<td>Dynamic environment</td>
</tr>
<tr>
<td>Optimization</td>
<td>Adaptation</td>
</tr>
<tr>
<td>Knowledge base</td>
<td>Behaviour base</td>
</tr>
<tr>
<td>Complete information</td>
<td>Incomplete information</td>
</tr>
<tr>
<td>Rational agent</td>
<td>Bounded rational agent</td>
</tr>
</tbody>
</table>

Figure 4.2: Deterministic vs. emergent approach (Ueda, 2007)
Part III

Results
Chapter 5

Test of standards for representing manufacturing system information

This chapter presents results and experience relevant for the first research question: How should resource information be modelled to be utilised for different purposes throughout the manufacturing system life cycle? Three hypotheses regarding the use of two STEP standards are tested using data from case studies.

5.1 STEP standards as a starting point

The initial hypothesis presented in Section 1.2 assumes that there is a need for an information model that can handle information from the whole manufacturing system life cycle as defined in Figure 1.4, in order to reach the research goal towards information integration. Due to the need to communicate information with various suppliers and systems, the starting point has been to find an already existing information standard that covers the information modelling needs of the manufacturing system life cycle.

In the following sections, hypotheses are formulated regarding the usability of the STEP standards AP214 and AP239. The hypotheses are tested using data gathered from case studies. The cases thereby exemplify real situations.

5.2 Test of STEP AP214

Based on the results from earlier research regarding the ability to use AP214 to represent information for manufacturing system development (see Section 3.3), the first approach was to test the usability of this standard also for the life cycle perspective. The defined scope of AP214 is information for the exchange between applications that support the development process of the mechanical aspects of automotive vehicles (ISO 10303-214, 2000). Although this implies that the standard is focused on
development, the standard’s applicability for representing manufacturing systems during its life cycle is first evaluated.

Hypothesis 1
The first hypothesis was formulated:

STEP AP214 is sufficient for representing manufacturing system information for the whole manufacturing system life cycle

This hypothesis was tested using data from the first case study.

Test case: Representation of PPR data of a machining line
The study object was a manufacturing system for engine block machining. The communication between the manufacturing company and its resource supplier was studied in order to capture what type of information was exchanged during development. The purpose of the study was to investigate if the necessary information for developing this machining system can be represented by the standard AP214, and if this representation holds over the life cycle of the manufacturing system. More details of this case study can be found in (von Euler-Chelpin and Kjellberg, 2004).

Modelling results and analysis
Figure 5.1 shows the PPR information structure for the studied machining line according to AP214, according to the same principle described in earlier research, e.g. (Mårtensson et al., 2002) and (Johansson, 2001). The instance in the figure is simplified: it shows only one process operation (Process_operation_occurrence), with relations to the resource used for this operation and the product to be machined. Products and resources can be modelled according to the same principle (Johansson, 2001), using Item, Item_version, Design_discipline_item_definition. The particular process operation defines what resource is used by pointing to a Single_instance of the resource definition (Item). There is also a relation from the process operation to the incoming or outgoing product which is defined as a Process_state of the final product (Item). Further, there is a relation from the Process_plan_version to the final product.

This model represents the manufacturing system from a development perspective. The representation shown in Figure 5.1 has some issues when the manufacturing system is regarded from a life cycle perspective. If the scope includes the operation phase, life cycle information such as maintenance, breakages, replaced parts, performance measures, status, etc, needs to be represented. This type of information cannot be represented explicitly by AP214 since it is not part of the scope of the standard. Life cycle data can however be put into documents, which can be associated to the PPR structure, but this makes information less computer-interpretable.
Figure 5.1: A simplified PPR model according to AP214, applied on a machining line of engine blocks, modified from (von Euler-Chelpin and Kjellberg, 2004)
Another important issue concerns the life cycle stage of the resource. In the model shown in Figure 5.1, both the product and the resources are represented from a design perspective. During manufacturing system development, this perspective is not wrong since the manufacturing system truly is designed. However, the particular resources are not designed. This means that if resources are to be represented in more detail, e.g., its components and tools, there needs to be an alternative way to represent structures than from a design viewpoint.

Moreover, after installation of the resources, the manufacturing system model may also need to include instances representing the real physical resources. In this situation, if AP214 is to be used, one should consider the use of Physical_instance as shown in Figure 5.2. The relationship is_realization_of connects the Physical_instance to the Design_discipline_item_definition of the design structure. In Figure 5.2, such relations are shown both on the system level and the resource level.

However, the differentiation between design instances and physical instances is not enough. Before the manufacturing resources are purchased, the resources are
neither designed nor physical; they are just planned. Thus, it can be concluded that for the representation of resources from a life cycle perspective, it is necessary to distinguish not only between the physical and non-physical perspectives, but also between the planning and design perspectives, where planning concerns the usage and the design concerns the design of the actual resource. From a user perspective, only a subset of the design information is relevant, such as the data found in technical specifications. The assembly structures and detailed geometries of the resource are neither needed by the resources users, nor made public by the resource makers. Thus, what is needed is a way to represent the manufacturing resource as planned where only the design information relevant for using the resource is included.

In other words, referring back to Figure 1.3, the production planning phase of the product life cycle is not supported by AP214 on the single resource level, only on the manufacturing system level, since an assembly structure is not the most appropriate way of representing a resource during planning and operation.

It can be concluded that an information model for the manufacturing system life cycle needs to handle at least three different life cycle perspectives for resource instances. These are defined as follows:

- **Definition:** A resource instance “as-designed” is a simplified and user-adapted resource instance containing information specifying the resource as a final product that is not yet assigned for a specific use.

- **Definition:** A resource instance “as-planned” is a resource instance based on an instance as-designed, containing further information regarding its role and configuration in a process.

- **Definition:** Resource instance “as-is” is a resource instance representing a real resource that is a realisation of an instance as-planned, with additional information that reflects how the real resource has changed.

There can be several as-planned instances of an as-designed instance, e.g. when the same resource type is used for several operations, and there can be several as-is instances of an as-planned instance, e.g. when two resources of the same type and identical configurations are needed. The purpose of having as-is representations is to be able to differ resource individuals on the shop floor if they have developed different characteristics.

**Conclusion for Hypothesis 1**

Since AP214 does not support the planning and use perspective of a resource, and since it does not cover operational aspects such as maintenance and other life cycle data, Hypothesis 1 is falsified.
5.3 Test of combination of STEP AP214 and STEP AP239 (PLCS)

The first case study exemplified the fact that each standard has a limitation. It is often the case that one standard does not cover all information that needs to be represented. The STEP standard AP239 – Product Life Cycle Support (PLCS) includes usage-related data, such as maintenance. In this way, PLCS has a suitable scope since it addresses the whole life cycle, but AP214 excels in representing manufacturing aspects, such as process operation structures.

Next hypothesis is focused on the combination of different STEP application protocols (AP) in order to choose the best aspects of each standard.

Hypothesis 2

Hypothesis 2 was formulated as:

STEP AP214 and STEP AP239 can be combined in order to be able to choose the best parts of each representation

This hypothesis was tested using data from the second case study.

Test case: Representation of the detailed process of an assembly cell

The case study was focused on representing all process conditions necessary to make the control code for a welding cell. The study object is a welding cell of two robots controlled by a PLC (Programmable Logic Controller) in a fully automatic Body-in-White system within the automotive industry. The cell includes a fixture that is used to clamp the body part. Two different product models are produced in this system, which means that different programs are run depending on which product model enters the cell. This makes it important to be able to represent alternative processes. Moreover, two robots work in parallel which makes it important to represent process conditions to avoid e.g. collisions. More details about this case study has been described in (von Euler-Chelpin et al., 2004).

The approach was to survey the information needs for making the control code, and to represent this information using the standard AP214 with additions from AP239. AP214 was chosen due to its convenient PPR structure and process entities. AP214 does not provide representation for the actual control code or behaviour models (Falkman et al., 2008) but most of the information needed to specify the control code can be explicitly represented. However, AP214 is not able to capture all aspects of the process control. For this reason, entities from AP239 were used to express different states of machine components and other conditions that control the execution of the process. This is described below.
Modelling results and analysis

Representation of process conditions using AP214 and AP239

Earlier, it has been described that AP214 allows resources to be assigned to a process operation in the role of executing the operation. However, in this case study, there is a need to express that some resources constrain the execution of an operation. There can be many types of conditions before a process operation can execute, involving both the process, the products and the resources. Examples are conditions that the preceding operation must have finished completely, that the incoming product is of a certain model, or that a resource has a certain state.

AP214’s possibilities to represent resources that constrain the execution of operations are limited. In the example of a required state of a resource, the entity Process_operation_resource_assignment has an optional attribute called Reason which can be used to describe the reason behind the use of the resource for a particular process operation. However, a string value such as “the condition is that resource X is in state Y” is less computer-interpretable than explicit modelling.

In order to express such conditions, the approach was to extend AP214 with entities from AP239 describing conditions and state: State, State_assignment, State_role, and Condition_assignment. By borrowing state and condition entities from AP239, the essential part of the process information needed for making control code can be described.

In Figure 5.3 modified from (von Euler-Chelpin et al., 2004), Process_operation_resource_assignment points out an instance of a fixture clamp with the id “Y14” as a resource participating in the particular operation, with the role “condition”. In order to show what type of condition it is, there is a Condition_assignment which points to a State_assignment on the fixture clamp “Y14”. The condition is that the fixture clamp with the id “Y14” has the required state “closed” for the operation.

There are also situations where conditions can be conditional, e.g. when a condition is dependent on another condition. In that case, Condition_assignment has been used to identify both conditions. For the sake of simplicity and convenience, it does not follow the condition representation in AP239 exactly. Condition_assignment is modified to point out the affected ends directly, like a relationship assignment. Another modification is to allow the Condition_assignment to point out multiple conditions. Figure 5.4 shows a situation where a Process_operation_occurrence has an “exclusiveness” relationship with another Process_operation_occurrence, which means the previous operation cannot be ongoing when the next operation starts to execute. However, this relationship only applies if the state of “Resource B” is at “work position and when the incoming product (Process_operation_input_or_output) is “Product A”. This situation is also called an interlocking specification, and has also been described in (Falkman et al., 2008) and (Falkman, 2005).

Another similar approach of representing of process conditions related to AP214
Figure 5.3: A combination of AP214 and AP239 for representing process control specification, adapted from (von Euler-Chelpin et al., 2004)

has been described in (Nielsen, 2003).

Conclusions regarding Hypothesis 2

The standards AP214 and AP239 can together cover a large part of the information needs for specifying control code requirements. Even though they are both generic standards with similar levels of abstraction, the combination introduces uncertainties since there are many ways to combine these standards. In this example, the standard AP214 was used as a base, and where it did not suffice, entities from AP239 were added. There are several other possible ways, e.g. using AP239 as a base with additions from AP214. This makes approach of mixing entities indistinct. Further, the addition of entities is a violation of the standard and should be avoided.

Hypothesis 2 can be corroborated in the aspect that the standards are theoretically combinable, but this approach is rejected since more promising alternatives have emerged. The next approach is to use AP239 and extend it with external classes defined in reference data libraries (RDL).
Figure 5.4: A model instance based on AP214 and AP239 describing a condition on an exclusiveness relationship (a not ongoing operation): one condition is on the incoming product and another condition is on the state of a resource

5.4 Test of STEP AP239 (PLCS)

Since AP239 (Product life cycle support) has a broad life cycle scope and is applicable for any kind of product, the next approach has been to explore the possibilities of using this standard. Since the experience from hypothesis 2 indicated that an arbitrary combination of entities from different standards does not seem appropriate, the possibilities to specialise AP239 with the use of reference data libraries (RDL) (see Section 3.3) was explored.

Hypothesis 3

Hypothesis 3 was formulated as: **STEP AP239 (PLCS) together with RDL is sufficient for representing and communicating manufacturing system information**

For this hypothesis, data from a machining line similar to the first case study is used. In brief, a part is machined by the use of machining centres, mainly through
CHAPTER 5. TEST OF STANDARDS FOR REPRESENTING MANUFACTURING SYSTEM INFORMATION

64

Figure 5.5: PLCS supports the distinction between a resource type (as-designed), a planned individual (as-planned), and a realised resource (as-is)

milling and drilling operations.

Modelling results and analysis

Life cycle stages

It has been concluded (see Section 5.2) that at least three life cycle stages of resources need to be distinguished: as-designed, as-planned, and as-is. PLCS allows this distinction by the use of the entities Part, Part_version, and Part_view_definition for design information, the entities Product_as_individual, Product_as_planned, and Product_as_individual_view for the development phase, and Product_as_individual, Product_as_realized and Product_as_individual_view for the operation phase. Figure 5.5 shows these variants, simplified from PLCS (ISO 10303-239, 2005).

Alternative breakdowns

It has been pointed out that all design information of a resource is not necessary from a user perspective. The components of a resource and their properties are of interest, but not how all components are assembled together. A few important dimensions such as length, width and some distances between moving components, etc, are relevant, but not the full CAD geometry.

A benefit using PLCS is the possibility to disregard the assembly structure of a resource and create other types of breakdowns, e.g. physical breakdown, system breakdown, functional breakdown, and zone breakdown. From a user perspective, it
is more convenient to represent subsystem elements as Breakdown_element instead of Part, since a relevant area of the resource from a user perspective sometimes represents a system or function rather than a mechanical element. Depending on viewpoint, different breakdown structures can be formed, e.g. one for the viewpoint of process planning, one for maintenance, etc. Figure 6.4 in Chapter 6 exemplifies further how a machining centre can be broken down into subsystems.

**PPR and process representation using AP239(PLCS)**

Compared to AP214, AP239 has a more ambiguous way of representing PPR structures. One of the reasons is that AP239 does not have entities specifically aimed for manufacturing processes. Instead, AP239 use more generic concepts, such as Activity and Activity_method. Figure 5.6 shows one example of how a process structure can be formed using entities related to Scheme. Scheme is a subtype of Activity_method and is used to describe an intended course of action to accomplish an objective, with the possibilities to order action entries (ISO 10303-239, 2005). Comparing AP239 entities to AP214, Scheme can be used for Process_plan, Scheme_entry can be used for Process_operation_occurrence, and Task_method can be used for Process_operation_definition.

However, there are other possibilities since Activity_method has more subtypes than Scheme. Due to several alternatives of how to represent processes, the use of PLCS to represent processes is prone to instantiation inconsistencies. The current DEXs do not currently give any guidance regarding how to represent manufacturing processes (OASIS PLCS TC, 2007).

Due to many instantiation possibilities, the utilisation of RDL classes to specialise entities into process-related concepts has been considered. If e.g. the process representation of AP214 is considered appropriate, these could be be defined as reference data classes. In other words, external classes can be defined as Process_plan, Process_plan_version, Process_operation_occurrence and Process_operation_definition. Since the external RDL classes bring meaning to the entities, the generic entity Activity_method which is a supertype for e.g. Scheme, Scheme_version, and Scheme_entry, could be used without further specialisation. This idea is shown in Figure 5.7. Similarly, the generic Activity_method_relationship can be used for relationship entities and be classified according to externally defined classes such as Process_operation_occurrence_relationship and Process_plan_relationship.

This approach shows the possibility to place most of the interpretation outside the PLCS model, and thereby make the exact choice of PLCS entities less important. Such classifications can also be applied to the variant shown in Figure 5.6 in order to add more meaning to e.g. Scheme. Moreover, this could be a more promising approach to combine AP214 and AP239 than the mixed approach proposed in Hypothesis 2. This would allow the utilisation of AP214 entities, but the structure would follow the AP239 model.
CHAPTER 5. TEST OF STANDARDS FOR REPRESENTING MANUFACTURING SYSTEM INFORMATION

Figure 5.6: Example of a process structure represented using AP239 (PLCS)
5.4. TEST OF STEP AP239 (PLCS)

In order to form a PPR structure, the relationships between an activity (process) and the used resource and produced product need to be established. A possible PPR structure (with the use of Scheme for process representation) is shown in Figure 5.8. The use of AP239 has the benefit of being able to represent resources from many life cycle stages: a certain resource model (Part), planned instances (Product-as_planned), and physical instances (Product_as_realized). In this figure, an as-planned instance has been chosen.

Conclusions regarding Hypothesis 3

The product life cycle approach of AP239 offers many suitable constructs, e.g. the differentiation between resources as-designed, as-planned, and as-realised (asis), as well as alternative breakdown structures. It can be concluded that AP239 can represent PPR information and process structures and thereby it fulfills the
CHAPTER 5. TEST OF STANDARDS FOR REPRESENTING MANUFACTURING SYSTEM INFORMATION

Figure 5.8: Example of how to represent a PPR structure using AP239 with the possibility to assign resources from different life cycle stages (here as-planned)

basic needs for manufacturing system development. Moreover, AP239 supports the representation of information from the operation phase such as maintenance.

However, the activity representation of AP239 is more generic than the process representation of AP214 which makes the instantiation more open and thereby more inclined for inconsistencies. Inconsistent instantiation is something that affects both AP214 and AP239, and perhaps all information standards. However, by classifying AP239 entities according to external classes for process concepts such as Process_plan, Process_operation_occurrence (taken from AP214), the exact choice of entities may lose significance, since the meaning of generic entities can be distinguished regarding how they are classified.

The ability to use RDL, as compared to STEP application protocols that do not use RDL, also reduces the issue of applying a consistent naming convention, which otherwise is important when using e.g. AP214 (Nielsen, 2003). For example, if Part is identified by a attribute value such as “Machine tool”, small spelling variations such as “machine tool”, “Machine Tool” or “Machine_tool” could cause interpretation problems. By classifying a Part as an external class Machine tool, the naming and identification problem is diminished.

To conclude, hypothesis 3 can be corroborated in the aspect that PLCS together
5.5. **REGARDING CONSISTENT INSTANTIATION**

With RDL is sufficient regarding the information scope, but with some remarks:

- There needs to be further guidance regarding how to consistently represent manufacturing processes (e.g. in a DEX)
- A product-specific guidance needs to be developed regarding *what* should be included according to knowledge of the domain, e.g. what to include in a resource model. Such a domain model needs to catch relationships between external classes. From this aspect, reference data should not be limited to specialising AP239 classes but rather form an ontology model.

### 5.5 Regarding consistent instantiation

In previous sections, the problem of consistent instantiation of generic standards has been expressed. It is however unreasonable to believe that a generic information model should have the sole responsibility for a consistent use. There is always a need for agreements, supporting guidelines and meta-models that tell how to use a model for a specific application.

The need for instantiation guidance for PLCS has already been addressed with the development of DEXs (see Section 3.3) by OASIS. However, there is also a need for guidance on a more specific level, such as how to represent manufacturing processes. Since PLCS lacks entities specifically intended for such processes, there is a need for a recommendation or guidance regarding which entities to use.

However, as will be shown below, there can probably never ensure a 100% directing guidance, since the choice of instantiation is also a matter of taste. This also applies for AP214.

**Instantiation inconsistencies: The process example**

To exemplify this problem, the following example is used: Operation 1 directly precedes both operations 2 and 3 which are to be executed simultaneously, but they are both directly preceding operation 4. It can be expressed as:

```
1 seq (2 sim 3) seq 4
```

The main relationships between different operations according to AP214 are shown in Figure 5.9. This example can be modelled using relationships between pairs only, or by forming groups and use relationships between groups, as shown in Figure 5.10. They express the same type of relationships, but in slightly different ways. `Process_operation_occurrence` is abbreviated as “p:o:o” and `Process_operation_occurrence_relationship` is abbreviated as “p:o:o:r”.

Another instantiation issue is how to deal with redundancies as compared to a minimalistic modelling strategy. For example, one might add redundant relationships that are already implicitly understood. E.g., in the instance on the right side
CHAPTER 5. TEST OF STANDARDS FOR REPRESENTING MANUFACTURING SYSTEM INFORMATION

70

Figure 5.9: Process operations and possible relationships according to STEP AP214

Figure 5.10: Two different ways of modelling the same process operation sequence

in Figure 5.10, it is technically possible to add a sequence relationship between e.g. operation 1 and operation 2, or between operation 1 and operation 3. However, this is already implied by the sequence relationship between operation 1 and the group 2&3.

This issue also affects information that is “attached” to process operations. For example, if the process operations have conditions (see Section 5.3), it also has to be decided where to put these conditions. Three possible ways are shown in Figure 5.11. Process_operation_occurrence is abbreviated as P.o.o, and Condition_assignment is abbreviated as C.a. To simplify, two different conditions are merely labelled “A” and “B”. In the variant to the left, all conditions are put on the sub operations. The strategy in the middle of the figure is inspired by inheritance, which means that the common condition that applies for all sub operations are put on the parent process, i.e. the process group for operations 2 & 3. The third way is to allow redundancy. All required states are expressed for each sub operation, and the common requirements can be put in the upper-level operation as well.

Each of the alternatives has issues. An over-defined model is easy to interpret, but the consistency of the model will be harder to maintain. A minimalistic inheritance approach would have easier updating, but the model would be less interpretable since one would have to look at upper-level operations in order to get the full context.
Schenck and Wilson (1994) do not recommend redundancy in models and suggest the use of derivation rules. Such rules can express formulas where one value depends on another. This cannot be shown in the graphical form of EXPRESS – only the lexical computer-interpretable variant.

**Implementations for aiding instantiation**

The different ways to represent the same thing has consequences on implementations reading from such models. For example, the work of extending the PDI\textsuperscript{net} web client (see Section 3.3) to enable viewing of PPR information based on AP214, clarified that a consistent instantiation of the standard greatly simplifies the creation of application logic. A generic model such as AP214 and AP239 often means inconvenient implementation, since many different modelling variations are possible. In general, the more that is standardised and fixed, the higher is the potential for automation.

Although the purpose of implementation is to interpret information, another purpose of implementations is to control the instantiation in a desired way. Since no information model is likely to guarantee a total instantiation consistency regarding its use, implementations that handle the creation and interpretation of information will contribute to a consistent use of a model.

The most important characteristic of an information model should be its ability to represent all information of the desired scope. A model’s inclination towards inconsistent use should not be the basis for ruling out a model, since there is a possibility to minimise such issues through the implementation of directing templates and converters. It is however important to be aware of that the use of very generic models such as AP239 needs more guidance than specific models, which forces the development of an extensive surrounding mechanism that control instantiation and interpretation. Thus, PLCS’s benefits in terms of a wide scope and high applicability is also its weakness, since it implies the need for more guidance.
5.6 Summary regarding the experience of using STEP

- In the choice between AP214 and AP239 to represent manufacturing system information from a life cycle perspective, the choice is AP239 due to its ability to represent the three desired life cycle stages (as-designed, as-planned, and as-is). Moreover, AP239 has alternative breakdown structures which is useful for representing a resource in detail, but from a user perspective.

- However, the risk of instantiation inconsistencies from using a generic standard brings about the need for further instantiation guidance. When the uncertainties concern what entity to use, the use of specific reference data classes (RDL) to classify generic PLCS entities is a promising approach. For example, process entities from AP214 can be used when defining reference data for manufacturing processes.

- The use of RDL does however not solve all instantiation problems. The current RDL developed within PLCS does not focus on defining relationships between the reference data classes. When representing manufacturing information, one would like to conform to general knowledge regarding how different manufacturing concepts are related. In other words, the instantiation guide should be another model with specific concepts belonging to the manufacturing domain. Such a model should function as a guide telling what information should be represented. In Chapter 6, a domain model of a machining centre is defined for this purpose.
Chapter 6

A resource model based on AP239 (PLCS) and a concept model

This chapter elaborates the representation of a machining centre using a concept model as a reference model for AP239.

6.1 The need for resource models

From the use of the STEP standards AP214 and AP239 in Chapter 5, it has been concluded that AP239 (PLCS) is the STEP standard that has the most suitable scope when the manufacturing system is to be represented from a life cycle perspective. However, a generic standard such as PLCS does not guarantee a consistent instantiation.

The mechanisms of reference data libraries (RDL) allows entities of the PLCS standard to be classified according to external classes. However, reference data libraries as they are used today do not provide guidance regarding what and how a resource should be instantiated. Neither do the current data exchange sets (DEXs) developed within OASIS, since they rather focus on the use of PLCS regardless of what product is being described.

For this reason, the approach is to create a domain model of a resource in order to support the instantiation of PLCS for a specific type of product – a manufacturing resource. The purpose of defining a domain model of a resource is to:

- Identify the preferred terminology for easier communication between computer applications as well as humans
- Express knowledge about a resource and its functions
- Express what information is needed to handle and provide a template for e.g. how to specify a resource
CHAPTER 6. A RESOURCE MODEL BASED ON AP239 (PLCS) AND A CONCEPT MODEL

- Function as a guidance of how to consistently instantiate a resource in neutral communication formats such as AP239 (PLCS)

The following sections will focus on what to represent and how to represent it using the modelling mechanisms of PLCS.

**Defining what to represent in a resource model**

The design of a resource model should start by asking what questions the model should be able to answer (see page 26). The questions depend on whether it concerns a resource type, a planned individual, or a physical individual. Examples are given below:

- General questions about the resource type (as-designed):
  - What kind of resource is it? (e.g. a machining centre)
  - What are its physical attributes? (subsystmes and properties)
  - What can it do? (e.g. supported processes, number of axes, tool interface)
  - How well can it do it? (e.g. capability information)
  - Where can it be placed on the shop floor? (e.g. media interfaces, dimensions, weight)

- Questions regarding resource usage (as-planned):
  - What is it planned to do (e.g. operation 1 in process A)
  - How is it configured? (e.g. which tools, which fixtures)

- Questions regarding resource status, performance, behaviour and history (as-is/as-was):
  - What is it doing? (e.g. operation 1 in process A)
  - How well does it do it? (observed capability)
  - What has it done before? (e.g. operation 1 in process B)
  - How well did it do it before? (observed capability)
  - What is its condition? (e.g. maintenance, repair)
  - How should it be maintained and operated? (e.g. operation manual, runtime experiences)

Not all these questions are addressed in detailed within this work. However, it is obvious that the resource model must contain more details than the black box level shown in previous models in Chapter 5. In other words, a resource model for the whole manufacturing system life cycle needs to handle the system within the resource as well as the system it is part of.
6.2 Concept model of a machining centre

A machining centre is chosen as an example since it is a flexible machine tool. It is a middle course from the alternatives of representing a more generic machine tool model, or a more specific machine tool such as a drilling machine. A machining centre has been defined as “a numerically controlled machine tool, where the spindle orientation is usually either horizontal or vertical, capable of carrying out two or more machining processes (e.g. milling, drilling, boring) and having facilities to enable tools to be changed automatically from a magazine or similar storage unit in accordance with the machining program” (SIS, 2001). From this definition, it is implied that spindles are used for the machining. Turning operations are thereby excluded from the domain model of a machining centre.

Model example: a machining centre

In the work of defining a concept model of a machining centre, four particular viewpoints have been taken into consideration: Process planning, factory planning, resource investment, and improvement work. These viewpoints correspond to the core development processes of the project ModArt (see Section 3.5). In reality, these processes are intertwined with many overlapping activities. In this work, however, they are separated as follows:

- Process planning involves the machining preparation process, which includes selection of operations, tools, and control code creation
- Factory planning includes design of factory layout, flow simulation, and installation
- Resource investment includes requirement specifications, evaluation of quotations and follow-up
- Improvement work includes monitoring and measuring the production in order to maintain performance and to find improvement potential

A first outline to map out relevant information concepts of a machining centre has been presented in (von Euler-Chelpin, et al., 2007), where partial concept models have been defined as below:
CHAPTER 6. A RESOURCE MODEL BASED ON AP239 (PLCS) AND A CONCEPT MODEL

- Concepts for process planning include the resource’s parts and properties, (e.g. table, axes, work zone) for the selection of resource type; interface information for the selection of tools and fixtures; process liquid usage; and control system information for the creation of NC code; and the process operations the resource is about to perform.

- Concepts useful for factory planning include outer dimensions of the machines for the layout, media interfaces and interfaces to external equipment for the installation, and various process times such as availability, cycle time, scheduled maintenance, tool change time for flow simulations.

- Concepts for operations/improvement include logged parameters and events such as accuracy, process capability, measures related to OEE, disruptions, and maintenance. Moreover, it includes other types of life cycle data indicating wear status.

- The viewpoint of resource investment includes many of the process planning concepts. In fact it may include anything that one would like to evaluate and specify as requirements. Aspects that do not concern the actual manufacturing process (e.g. ergonomics and safety) are however not considered.

Machining centre concepts from these viewpoints are unified in Figure 6.1 which has been developed in collaboration with members\(^1\) of the ModArt project. It has been based on information concepts found in computer-aided process planning tools, flow simulation tools, requirement specifications, quotations from resource suppliers, and the standard ASME B5.59-2. Concept models for separate viewpoints can be used to filter the total amount of resource information to show only the relevant aspects.

It can be noted that not all of the concepts are typical resource concepts, such as process concepts. However, since related concepts form the context of a concept (Naeve, 2001), these need to be included since it would be difficult to give a good description of a resource without mentioning the process and the produced product. The continuity of concepts makes it difficult to make a delimitation of a concept model. The more that is included, the more it becomes a concept model for manufacturing rather than for a resource. The scope of the concept model in Figure 6.1 is however adapted to suit the questions and viewpoints listed above.

The concept model shown in Figure 6.1 is not comprehensive and for the sake of readability it does not show all possible relations and concepts. In the end, it is desired that the chosen terms are as recognised as possible, and that also the relationship types are limited to recurrent types (“has”, “has property”, etc). To clarify the concept model further, each concept and relation should be enriched with a textual definition as exemplified in Figure 6.2.

\(^1\)Mikael Hedlund in particular
Figure 6.1: A non-exhaustive concept model of concepts related to a machining centre, shaded according to generic information types
This concept model can be extended to represent other types of machine tools, such as machines for milling and turning. It has however little in common with other types of resources such as robots and conveyors.

### 6.3 Using PLCS and a concept model for representing a machining centre

In Chapter 5 it was concluded that the STEP standard AP239 Product life cycle support (PLCS) is suitable for representing resources from a life cycle perspective. Some of the strengths of PLCS compared to AP214 is that it supports more life cycle stages and that it has several breakdown opportunities for structures. This section presents how PLCS can be used together with the concept model described in the previous section.

#### Representation of resources using AP239

Figure 6.3 summarises what parts of the PLCS model can be used for representing different aspects of a resource. Each aspect is explained below.

**Life cycle stages**

The representation of resources has previously been described in Section 5.4. The necessary life cycle stages that have been identified, i.e. as-designed, as-planned, and as-is, (see Section 5.2) can be distinguished by the use of `Part_version`, `Product_as_planned`, and `Product_as_realized`, respectively (see also Figure 5.5).

**Breakdowns**

Since a design structure is not relevant from a user perspective, other types of breakdowns can be formed, such as `System_breakdown` and `Functional_breakdown`. Figure 6.4 exemplifies how a machining centre can be structured into two
6.3. USING PLCS AND A CONCEPT MODEL FOR REPRESENTING A MACHINING CENTRE

<table>
<thead>
<tr>
<th>Information aspects:</th>
<th>Suggested PLCS representation:</th>
<th>Purpose:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource – as-designed Resource – as-planned Resource – as-is</td>
<td>Part, Part_version, etc Product_as_individual, Product_as_planned, etc</td>
<td>Life cycle stage</td>
</tr>
<tr>
<td>Components and subsystems</td>
<td>Part, etc Breakdown (System/Functional/Zone/...) Breakdown_element (System/Functional/...)</td>
<td>User’s viewpoint</td>
</tr>
<tr>
<td>Properties of the resource and its components</td>
<td>Assigned_property, Property_representation, etc</td>
<td></td>
</tr>
<tr>
<td>Processes and control</td>
<td>Activity, Activity_method/Scheme/Task_method State, Condition, etc</td>
<td></td>
</tr>
<tr>
<td>Runtime data</td>
<td>Event, Activity_actual, Observation, Property_representation, Document, etc</td>
<td></td>
</tr>
<tr>
<td>Domain-specific concepts</td>
<td>RDL, Classification_assignment, etc</td>
<td>Consistent terminology</td>
</tr>
<tr>
<td>Geometry, and other details</td>
<td>Document, Document_assignment, etc (+ other standards)</td>
<td>Outlying details</td>
</tr>
</tbody>
</table>

Figure 6.3: Suggested use of the PLCS standard for representing a resource breakdowns, one System_breakdown and one Functional_breakdown. The system breakdown has one system element (Spindle system) and the functional breakdown has one functional element (the X axis motion). In order to associate the X axis motion with the spindle system, there is a Breakdown_element_realization relationship between these elements. In order to associate the spindle system element to a corresponding Part, Breakdown_element_realization can be used. Thus, parts and components can be related to the machining centre via breakdown structures without bothering about how they are physically assembled. Moreover, the use of a breakdown element instead of Part can be used to signal that the component in question is regarded from a user perspective rather than from a design perspective.

Properties

Properties are attached to the Part_view_definition associated to a Part, and similar constructions are used to its subtypes. Thus, properties can be assigned to a Breakdown_element_definition associated to a certain Breakdown_element, where Breakdown_element is used instead of Part.
Figure 6.4: Simplified example of a system breakdown and a functional breakdown of a machining centre according to AP239 (PLCS)
Processes

One possible way of representing process structures has been described in Section 5.4 using entities related to Scheme. In cases where the process needs to be expressed with e.g. interlocking conditions, State and Condition can be used related to process operations and resources as described in Section 5.3.

Runtime data

Runtime data can e.g. be explicitly represented in the form of an Event, a Property_representation (associated to a Product_as_realized, see also Figure 7.6 in Chapter 7), an activity that has been carried out (Activity_actual), or as an Observation. Runtime data can also be represented inside a Document.

Using the concept model as a reference model for PLCS

One of the advantages of using RDL is the ability to reduce the significance of the attribute values of entities for the role of identifying what type of information it represents. In other words, instead of an attribute value named “machining centre” to identify what kind of item it represents, there can be a classification to an external class with that name. This will decrease the instantiation inconsistencies that are caused by spelling variations.

For this reason, the proposed approach is to classify entities using external classes according to the concept model described above, in order to use a consistent and recognised identification of objects. Using process representations as an example, generic PLCS entities such as Scheme or Activity method can be classified as a more interpretable class Process operation. This principle has been shown earlier in Figure 5.7.

Mapping concepts to PLCS

The approach is not only to use the concept model to classify PLCS entities, but also to create a PLCS structure. An initial mapping to PLCS has been illustrated in Figure 6.1 by shading the concepts according to a group of entities in PLCS. The details of using the concept model as an extended instantiation guide is elaborated below.

Figure 6.5 extended from (von Euler-Chelpin, et al., 2007), shows a piece of the concept model where two types of concepts can be discerned: machine parts and their properties. The relationship between machine parts is “Has” and the relationship from a machine part to a property is “Has property”. A machine part such as “Spindle” can, depending on viewpoint, be represented either by Part_version, Product_as_planned, Product_as_realized, or as a breakdown element, such as System_element and Functional_element.

Properties such as “weight” are represented using Assigned_property, Property_representation, etc. The relationship “has property” in the concept model
CHAPTER 6. A RESOURCE MODEL BASED ON AP239 (PLCS) AND A CONCEPT MODEL

Figure 6.5: There are more than one way to map concepts and relations to PLCS entities

can be associated to the relationship described_element, which connects properties to e.g. Part_view_definition or System_element_definition.

The relationship “has” in the concept model implies a structure. There are several ways to form structures. System elements can be decomposed using the System_element_usage relationship. If a system element is associated to a part, the Breakdown_element_realization relationship is used. If the system element is directly under the top of a breakdown, the relationship System_breakdown_context is used, pointing to a System_breakdown_version (of a System_breakdown), and then associated to a Part_view_definition via Breakdown_of.

A functional aspect of a resource, such as the Z axis motion of a machining centre, can be represented as a Functional_element which can be realised by one or many System_elements or Parts. Since properties of an axis can change according to what system element realises it (e.g. a Table or a Spindle head), properties can be associated to the combination of a functional element and a system element. Thus, properties can be assigned to the relationship entity Breakdown_element_
6.3. USING PLCS AND A CONCEPT MODEL FOR REPRESENTING A MACHINING CENTRE

Figure 6.6: Relationship between functional elements and system elements of a machining centre

realization instead, as shown in Figure 6.6.

As can be seen, there is not a 1:1 mapping between the concept model and the PLCS model but rather an n:n relation. Thus, even if a concept model is used as an instantiation guidance, a consistent instantiation cannot be achieved regarding which entities to choose for machine elements, where to put properties, etc, since the choice of PLCS entities depends on the life cycle stage, viewpoint, and decision stage. What complicates the situation even more is the fact that machining centre concepts can be associated to each other in more ways than shown in Figure 6.1. A concept model should not be designed to exclude additional relationships.

Pros and cons regarding the use of PLCS

Since PLCS offers so many ways to represent e.g. a machining centre, there could be situations where the exact place of relevant information can be hard to remember. In other words, finding relevant information could be an issue. For this reason, the ability to classify entities according to external classes named according to the terminology defined in the concept model becomes crucial. A consistent terminology
CHAPTER 6. A RESOURCE MODEL BASED ON AP239 (PLCS) AND A CONCEPT MODEL

used for classification reduces the significance of the chosen PLCS entities and relationships, since the reference data classes function as keywords for information search. The concept model also provides a way to get a quick overview of what types of information that is handled.

So far, it has been assumed that all information could be represented using PLCS. However, there is information that can be considered too detailed or that does not have the requirement of being treated from a life cycle perspective, e.g. 3D-geometry and detailed kinematic movements. Since there are more specific standards for this type of information within the STEP family, those standards should be used instead. Further, it is always possible to embed such information as documents in the PLCS model, e.g. in Part 21 format.

Another situation where the use of PLCS can be questioned is the case when the need to reuse and communicate information is limited. Generic standards are appropriate when the integration spans over a large integration scope. For communication and reuse within a narrower scope, simpler and more specific integration formats should be more suitable.

A drawback of choosing many different communication formats is however that a heterogeneous model environment requires more translations between information models. However, the choice of using PLCS solely has other drawbacks. The flexibility of PLCS makes it possible to represent close to everything, but the size and complexity of the model will also grow accordingly. Moreover, a lot of instantiation guidance is needed. The pros and cons of using PLCS as information model needs to be weighed up for each information type and situation.

Issues regarding the concept models

There are also issues regarding the development of concept models.

Since concept models are rather detailed and more specific than most information models, a consensus on the model can be an issue. Even within a manufacturing company there are probably enough differences in viewpoints to make it difficult to agree on one fixed concept model. An alternative to agree on a fixed concept model is to agree only the terms. One could compare it to using concepts as pieces that can be related according to one’s needs, as long as it does not conflict with the concept definitions. Changeable concept models with movable relationships would make it even more difficult to use it as an instantiation guidance.

If, however, concept models can be agreed on, there is a possibility to represent such models in OWL format (W3C, 2004) and be used as a reference data model for PLCS. Development of more domain-specific DEXs such as “representing a machining centre” is a possibility.

The question is however if the gap between the exemplified concept model of a machining centre and the PLCS model is too large when it comes to abstraction level. The multiple instantiation variations shown in e.g. Figure 6.5 and Figure 6.6 implies that there cannot be only one way of translating a concept model to PLCS. To diminish such a gap, the use of intermediate standards should be
considered. The upcoming machine tool standard ASME B5.59-2 can potentially function as an intermediate standard.

6.4 Summary regarding resource models

- The PLCS standard can handle most information aspects of a machining centre, e.g. alternative life cycle stages and breakdown structures, and is thereby a standard that has potential to capture all desired aspects of a resource during its life cycle.

- A model with many possibilities (such as PLCS) is also a model with many possibilities to do things differently. The use of RDL for classifying entities according to common information concepts has been stressed many times. However, RDL alone does not ensure a consistent instantiation.

- A concept model of a resource further tells what information is relevant to model, and could thereby be used as a template for machine specifications.

- By dividing the concept models into specific viewpoints such as process planning, factory planning, improvement, it can be defined what information is relevant for whom. PLCS is a format for how to model information, but there is also a need to direct how to apply PLCS.

- Due to ambiguous ways to map resource concepts to PLCS entities, the consistency problem cannot be eliminated by concept models alone. The use of more specific standards as intermediate integration levels can be considered.
Chapter 7

Runtime data and feedback from operation

This chapter addresses research questions 2 and 3, which are “How can runtime data be used to ensure that there is an accurate model of the manufacturing system?” and “How can experience from the operation phase be utilised in the work processes of all life cycle phases?”

7.1 Using runtime data

So far, the focus has mostly been on the manufacturing system from a development perspective, with resources as-designed or as-planned. This chapter focuses more on the representation of real resources and information generated during the operation phase. The term runtime data has been used to comprise all data that can be gathered during operation related to the production (see page 46).

The second and third research questions are closely related. Having an accurate model of the realised manufacturing system’s current state and runtime history provides opportunities for using this information, both for improving the production and for feeding back information to the development phase. The intended scenarios of improving the operation and development activities can be of various kinds, e.g.:

- How can key performance indicators such as availability, quality rate and performance rate be improved?
- Which resource has the most serious quality problems?
- What are the common problems and error types of a resource individual, and can these problems be avoided?
- What is the suitable maintenance frequency of a particular resource?
- How can the cost of operating a particular resource be decreased?
• What is the most common cause of disruptions?
• Are there product features or tool types that are error-prone?
• Is there a resource individual with quality problems?

Runtime data types
In order to understand the different types of runtime data that can be gathered, a rough distinction has been made between three general types of data: events, parameters and experiences (von Euler-Chelpin et al., 2006). In this categorisation, an event is a label for anything that can occur that is of relevance for evaluating resource and process capability, e.g. machine alarms, disruptions, and activities such as executed maintenance, inspections, etc. Parameters include everything that can be measured or calculated during runtime. It can e.g. be data related to the resource, process or the produced product, or the manufacturing environment. The term experience has been used to label a synthesised form of runtime data, such as results from analyses of various kinds. Examples are alarm-cause-action analyses and improvement analyses.

Another categorisation can be made regarding what the runtime data concerns. Some runtime data concerns the resource (e.g. alarms, failures, vibrations), the process operations it executes (e.g. cycle time, processing time), and the product it produces (e.g. quality). However, for some properties, it is difficult to decide where it belongs (e.g. MTBF which is almost as related to the process as it is to the resource). In fact, all runtime data are more or less associated to the whole manufacturing context.

Yet another distinction of runtime data is how synthesised and interpretable it is. In this case, raw data is the most unsynthesised form of data, such as logs of data of various types. Thereafter come measures of different types, e.g. availability, quality rate, and MTBF. Such measures are easier to interpret than raw data (compare e.g. an availability measure to a log of disruptions). Another step of synthesis corresponds to what has been called “experience” above. An experience can be e.g. an analysis or a problem case. Such information is the most interpretable since it has the most context and interpretation rules.

The importance of context
Even if an experience is a more interpretable source of knowledge than a log of runtime parameters and events, an experience is always created within a specific context. All types of runtime data (or knowledge originating from runtime data) must be interpreted within the prevailing situation when the data was captured.

Figure 7.1 shows a concept model illustrating the main types of contextual information that is relevant to capture in order to interpret data correctly. Runtime data has two important associations: to what it concerns (what it is a measure of) and to the context within which the data is valid. The figure indicates that it may
7.2 Runtime data for improving operation

When an improvement area is identified, e.g. improving capacity, the relevant information needed for an analysis can be gathered. Information can also be gathered to store cases to be used as a reference for similar cases in the future. One fictive example of such an application is presented below.
Gathering cases to improve operation: the alarm example

This example uses machine alarms as runtime data. By associating machine alarms with causes, symptoms, actions, etc, alarm cases can be created, which could function as a basis for automatic suggestion of actions to future alarms. Figure 7.2 shows information required for analysing machine alarms – why they occur and how to act, illustrated as a concept model.

This scenario is set on the local level, i.e. it concerns a specific resource individual, but it uses both local knowledge and general knowledge. The general knowledge is the alarm list provided by the resource supplier, specifying appropriate actions when different alarms occur. Local knowledge is the gathering of cases valid for a specific machine individual.

The general knowledge, i.e. the alarm list, can be formalised into logical rules. Examples of such rules are written below:

\[
\begin{align*}
\text{alarm}(X) \land \text{observation}(O) & \rightarrow \text{cause}(A) \\
\text{alarm}(X) \land \text{observation}(P) & \rightarrow \text{cause}(B) \\
\text{cause}(A) & \rightarrow \text{action}(V) \\
\text{cause}(B) & \rightarrow \text{action}(W)
\end{align*}
\]

where

\(X = \text{“SPINDLE INHIBIT”}\)
\(O = \text{“Chuck not closed”}\)
A turning spindle rotation command has been given when the chuck of the spindle was not closed

V = “Close the chuck of the spindle”

P = “Clamping device not clamped”

B = “A turning spindle rotation command has been given when the clamping device at the end of the milling spindle was not clamping”

W = “Clamp the clamping device at the end of the milling spindle”

This scenario is described in Figure 7.3, and a test implementation has been made using a Java program that interacts with a Prolog source using JPL (Java Interface to Prolog) (JPL, 2002). The facts and rules about alarms, causes, and actions are written in Prolog format. A goal-driven rule-based expert system algorithm in Prolog format taken from Luger (2002) is used to reason about these facts and rules, and where observations are expected the algorithm asks for answers (yes/no). The algorithm calculates the probable cause of the errors using certainty factors, and can thereby suggest an appropriate action.

Earlier alarm cases are stored in a database. Thus, it is possible for the program to use the database to get e.g. the most recent executed action to this problem or the most common executed action and thereby enrich the general information with real cases from the past. After an action has been identified and executed, the user is urged to write the executed action for the current case. Thus, a new experience is completed which can be added to the case base. This idea has also been described in (von Euler-Chelpin and Kjellberg, 2007).

This example shows that there can be local knowledge and general knowledge, where the local knowledge can be used as a complement to the general knowledge. The local knowledge is formed by synthesising runtime data valid for that specific machine (see Figure 7.4, modified from (von Euler-Chelpin et al., 2006)).

The additional local knowledge has of course a limited reusability since the
CHAPTER 7. RUNTIME DATA AND FEEDBACK FROM OPERATION

Figure 7.4: Acquiring runtime knowledge as a complement to general knowledge

gathered experiences are only relevant for the particular resource individual and its current configuration. It might not be applicable if the manufacturing context changes.

7.3 Runtime data for feedback to development

A greater challenge regarding the use of runtime data is how to feed back experience to development activities, and relate runtime data to the more static nature of development information. It has been mentioned that the context is important, and that raw data should be synthesised to key measures, experiences, and analysis results, in order to increase interpretability. Still, the reusability of runtime data is limited due to the dynamic environment during operation. When the environment changes, data can be misleading.

However, the integration of information from the development and operation phases would be facilitated if the feedback scenarios are known, i.e. scenarios defining what kind of runtime data is useful when and for what purpose. By knowing the receivers of operation feedback, the as-is representation of the manufacturing resources can be prepared with the focus on those purposes. In Section 3.5, the need for process or activity models was stated, since they describe what type of information is needed for what activity during the manufacturing system life cycle.

Receivers of runtime data

The use of runtime data for feedback to development depends on the type of development scenario. Four main scenarios can be discerned (von Euler-Chelpin, et
7.3. RUNTIME DATA FOR FEEDBACK TO DEVELOPMENT

al., 2007):

1. Old resources for old product: e.g. improvement work
2. Old resources for new product: the existing system is reconfigured for a new product variant
3. New resources for old product: resource investment for improving capacity or quality
4. New resources for new product: a new product version requires completely new operations and new resources

There are of course variants and mixtures of these situations. Moreover, a “new” resource could range from being a completely new resource type to being another individual of the same type and brand. Among these scenarios, the situations where resources need to be reused will be compared to the situations where new resources need to be invested in.

Reusing resources vs. investing in new resources

Figure 7.5 shows two scenarios: one where an old resource is reused for new operations and one where a new resource is needed.

In the cases where resources are reused, runtime data from real resources can be used for e.g. process planning and flow simulation. The original specification of a resource can be enriched with indications regarding the wear and status of the real resource by taking part of common error types, stability and accuracy deviations, etc. Indications about its wear status can also be estimated through executed repairs and maintenance. For the flow simulation, process parameters can be estimated with support from real values, e.g. MTBF, MTTR, and setup times.

In the cases where new resources are to be purchased, there are no runtime data available. However, data from similar resources (if any) can be used to estimate process times for flow simulations. During the warranty and follow-up phase of the investment process, runtime data can be used to see if the purchased resource fulfills the specified requirements. Experience from continuous use can also be used for defining requirements for future investments.

Comparing expected values with observed values

Elaborating the situation of follow-up during warranty time, a resource’s expected capability can be compared to the real measured capability. According to wide definition of capability on page 48 in Section 4.2, resource capability includes properties such as the quality of the produced output and the measured cycle time. Values of those properties have different meanings depending on the life cycle context. Properties during the design phase often describe boundary values (e.g. maximum spindle speed), properties during the planning phase often describe expected values
(e.g. expected cycle time), and properties during the operation phase represent observed values (e.g. actual cycle time) (Holmström and von Euler-Chelpin, 2004).

Figure 7.6 shows how the distinction of the meaning of a property can be derived from the life cycle context of the resource to which the property is assigned. The figure is based on AP239 (PLCS): The life cycle stages of a resource are distinguished using the entities related to the resource as-designed, as-planned and as-is (see also Figure 5.5 in Chapter 5). The meaning of a property value changes accordingly.

Further distinction of the property context might however be necessary if it is desired to express the exactness of a property or the origin of a property. For example, expected properties can be based on calculations (e.g. cycle time based on geometry, spindle speed, etc) which usually are more exact than properties based on estimations and guesses (e.g. MTBF of a new machine).

**Requirements for runtime data feedback**

The described scenarios indicate the need to maintain a representation of a real physical resources (i.e. an as-is instance) on which relevant capability information can be found, evaluated, and compared.
An issue is however how runtime data should be formatted. As has been mentioned in Section 7.1, runtime data can be synthesised in various degrees where unsynthesised raw data usually is the least interpretable. It is likely that the appropriate format of operation information depends on purpose and the receiver. For example, the requested format could be an instant value, a log of instant values from a certain time period, mean values or mathematical distributions based on data from a certain time period, a grade on grading system, an evaluation report, or perhaps oral feedback.

In reality, the feedback that runtime data can bring probably has a limited significance during manufacturing system development. Many decisions regarding the selection of resources are likely made by tradition, experience and other intangible factors. However, measures to increase the accessibility and interpretability of runtime data would increase the influencing opportunities. The access of relevant information is the basis for knowledge and competence of making good decisions (see Section 3.1). Some important factors to increase the availability and interpretability of runtime data has been identified:

Figure 7.6: The life cycle stage of the resource determines the context of properties (example using AP239/PLCS)
• The presence of an as-is-representation of resources where important operation information can be found. Properties and documents describing runtime data can be associated to the as-is representation and thereby be retrieved for comparing expectations with the outcome, or for acquiring wear indications.

• The specification of where and why runtime data should be used during development activities. For example, in the process pilots of the ModArt project (see Section 3.5), it should be specified in the activities where and what runtime data is useful, in order to increase the awareness of the feedback opportunity.

• The format and level of synthesis can also be crucial. Feedback from operation should be served in the format the receiver (human or system) prefers, whether it is raw data, measures, diagrams, text documents, or other formats.

• The access to necessary contextual data needed for interpreting runtime data.

7.4 Representation of real resources as-is

It has been stated that the presence of an as-is representation of resources is needed in order to access information from the operation phase. The second research question also implies that a model of the realised system should work as a surrogate for the real system. In such case, the model must give a trustworthy representation of the reality at all times. Further, the initial hypothesis (see Section 1.2) assumes that the representation of the resource as-is should be included in the same model as the resource models used for planning. This section discusses these ideas and what role an as-is representation should have.

Runtime data model vs. as-is representations

In the feedback scenarios described earlier in Section 7.3, the role of an as-is instance is to provide information from the operation phase in various formats. The exact need for information depends on what and how runtime data is to be used. It is obvious that the more information one would like to use, the more runtime information needs to be included in an as-is representation.

A detailed as-is representation would in turn put stronger requirements on the updating mechanism. The more details, the more needs to be updated if it shall remain a good model. If e.g. the model contains measures such as availability, quality rate, mean cycle time, etc, the model needs to be updated with a given time period.

The use of a generic model such as PLCS for representing runtime data would lead to a lot of overhead data. Even if there are cases where it is advantageous to have observed data in the same model as planned values, it is unreasonable that all runtime data be included in an as-is representation based on PLCS. For this reason, a separate model is suggested for the sole purpose of identifying and structuring the
runtime data needed for activities related to operation and improvement, today and in the foreseeable future. Due to the relative compactness of a relational database makes it more suitable for storing raw data, as well as various measures and analyses originating from this data.

The runtime data model should also focus on catching the data needed for rendering the manufacturing context as described in Section 7.1. It is desirable that there is a traceability possibility for a specific situation, concerning what feature was machined, what a machine individual was used, what process parameters were used, etc.

Regarding the update frequency, logged data with a fixed periodicity is important for the ability to interpret measures consistently and to analyse trends. The amount and frequency should be adapted to the needs, since the gathered data should have a purpose. However, since all possible analyses of runtime data cannot be predicted beforehand, a certain amount of “uninteresting” data should be justified since the lack of data would be a worse alternative.

So what should an as-is representation of a resource focus on? From the points above, a model instance representing the real physical resource on the shop floor should focus on information that is explicitly asked for, e.g., in a work process description, prepared in a format that suits the receiving system or person. An as-is model of a resource is thereby based on the as-planned model, enriched with runtime information synthesised into an appropriate format, e.g., a list of serious breakdowns, quality rate diagrams, OEE, etc. The interpreter of such information is likely to be a human, which makes a document-based modelling approach convenient.

Asking a model or asking the resource?

Having as-is instances that work as surrogates for real resources brings about questions regarding how the updating is supposed to be carried out. The more data that is associated to the as-is representation, the higher is the requirements on reliable updating. Each manual information management activity, such as typing in data, is associated with a cost. Naturally, operators should focus more on the real system than its representation. For this reason, information management of a repeating character should be automated as much as possible.

The increased computerisation of technical products in general (cars, photocopiers, etc.) leads to possibilities to let products maintain an as-is model themselves. Since sensors often are used to communicate to the user what is wrong, and to generate logs of various kinds, the idea of letting a resource update its own model should be possible. In this way, the current status of the resource as well as statistics of errors and repair could be fetched by asking the resource rather than by asking a user-designed model.

A consequence of such a scenario is that much of the responsibility of defining and updating an as-is representation will be put on the equipment manufacturers or control system manufacturers rather than merely being a concern of the user. It
would also be important to use a common terminology not only within an organisation, but also in a supplier network including resource manufacturers and software providers. In such a scenario, the concept model described in Chapter 6, should be based on an international standard.

In the future, there might also be more applications of decentralised control and management related to initiatives such as evolvable assembly systems (EAS) and biological manufacturing system (BMS) (see Section 4.4). Although, evolvable machining lines sounds a bit surrealistic, the ability of a machine tool to maintain a model of itself would at least provide opportunities for self-diagnosis and self-healing.

7.5 Summary regarding runtime data

- Runtime data can range from raw data to advanced knowledge representations of experiences. The more synthesised, the more interpretable is it likely to be.

- All data from the operation phase need to be interpreted within its context. The context is defined by the process context (including product, process and resource information) and the runtime context, which includes runtime data from the same measurement period as the data that is being interpreted.

- Gathered experiences can be used for improving and adjusting the production. The alarm example shows how appropriate actions can be found both using theory (when represented in a computer-interpretable format) and experience (old documented cases).

- When runtime data is used for feedback to development, the integration of information is an issue. The context is important, and the format and level of synthesis should be adapted to the receiving person or system. Moreover, the use of runtime data should be included in the development process descriptions. The availability of interpretable runtime data creates opportunities for better decisions.

- An as-is representation of resources makes it possible to e.g. compare observed values to the expectations. The as-is model needs to be regularly updated if it shall remain a good model. The more dynamic properties the model has, the higher are the demands for frequent updating.

- The large amount of runtime data and the unpredictable needs regarding future analyses leads to the need for a compact data storage. Thus, an as-is instance solely represented using PLCs is thereby considered inappropriate due to its voluminous format. A more suitable approach is to keep a separate runtime data model for the purpose of gathering relevant data on which various analyses can be based.
• Since information management during operation could cause extra work, model updating work of a repeating character needs to be automated to decrease manual administration. The trend towards intelligent products should make it possible to let the resource maintain a model of itself.
Part IV

Discussion and Conclusions
Chapter 8

A modelling discussion

From the preceding chapters, several challenges regarding information modelling have been experienced. The purpose of this chapter is to discuss the main challenges of modelling the manufacturing system from a life cycle perspective. The initial hypothesis is evaluated and an initial modelling approach is suggested.

8.1 Structuring a complex multi-dimensional world

Information modelling is associated to many choices. The modelling scope defined in this thesis, i.e. the manufacturing system during its life cycle, is a complex domain since it involves many time aspects, abstraction levels and a rich information content. As was experienced in Chapter 6, even a resource model has a considerable complexity if it is supposed to include all aspects relevant for the manufacturing system life cycle. In order to choose an appropriate modelling strategy, the main dimensions causing this complexity is first summarised.

Time aspects: Life cycle stage and information validity

The life cycle scope brings with it a complexity related to time. Figure 8.1 shows three life cycle stages: design, planning, and operation, where the model of each stage represents the resource as-designed, as-planned, and as-is. As has been defined in Section 5.2, an as-designed instance only includes information about the resource as a final product. Such information is usually found in technical specifications, including dimensions, weight, travel range of the axes, and spindle speed. In the planning phase, additional information regarding the resource usage is gradually created during manufacturing system development. Usage-dependent information includes assigned operations, fixtures and tools, and expected cycle time. During operation, runtime aspects can be added.

Along with the life cycle stages, the validity of information changes from being practically constant to momentary, as shown in the lower part of Figure 8.1. Data
describing the design intent of a resource can be considered constant. Information associated to its usage is variable, since the validity ceases if there is a change in configuration. The validity of runtime data gathered during operation can be considered momentary. Data indicating runtime behaviour and performance have short validity and seize to be valid during the next measurement or logging period.

The shift from constant to momentary leads to an increasing amount of instantiated data. For example, a planned resource has design information and usage information, and a realised resource has design information, usage information and runtime data. The amount of data does not affect the complexity of the model, but it may affect the models’ purposes. For example, the purpose of the models used during the development phase is different from the purpose of models of the operation phase:

- The purpose of models used during development is to represent something that does not yet exist, while the purpose of models used during operation is to represent something that does exist.

- During development, manufacturing system information gradually changes from being preliminary to fixed as decisions are made. During operation, the manufacturing system model becomes more and more approximate since the real manufacturing system is changing.

- During development, the real manufacturing system is designed according to the model. During operation, models should be adjusted according to the real manufacturing system. (If, however, the measures show a deviation from e.g. important tolerances, there should be an invocation to act on the real world, e.g. adjust the real resources, in order to meet the specified requirements defined in the planned models.)
Hierarchical aspects

There are also various hierarchical aspects to consider, including abstraction levels, detailing levels, and meta-levels.

Concerning abstraction levels, a “machining centre” can be placed in a resource classification as shown in the left part of Figure 8.2. Classifications can be created in respect to various qualities, such as process type, number of axes, or resulting feature. A continuation of this kind of classification upwards leads to e.g. resource, product, part, etc up to a universal concept.

The abstraction level is significant during modelling since the chosen abstraction level affects the captured characteristics of a resource. For example, a generic model is applicable for all types of resources and is less sensitive to changes in the real world. A more specific model, e.g. a milling machine model, has a relatively limited applicability, but more specific knowledge can be captured. Such a model can explicitly express that there needs to be at least one spindle and that milling operations are performed. In general, a generic model has wide applicability but contains less domain knowledge, whereas a specific model has limited applicability but is more directing and catches more knowledge of the domain. The combination of specific and generic aspects has been approached within the STEP community by the use of reference data libraries (RDL, see Section 3.3).

Another way to distinguish information is by its detailing level, which can be as much associated to a hierarchical dimension as with a time dimension. In Figure 8.2, a hierarchical structure is shown, where the different stages of detailing levels indicate how information evolves in the activities of a development processes. For example, in a development scenario, it is first decided that some machining centres are to be used. Information about resources as-designed are evaluated. When these machining centres are to be configured to specific operations, one would like to treat them individually, thereby creating instances as-planned, each with its own configuration. When machines are delivered, as-is instances can be created representing physical instances with unique serial numbers.

A third hierarchical variant is the meta-level, a level where the underlying level is “administered”. Such relations are usually established between models, rather than within one model. The generic standard PLCS can be used as an example, where DEXs guide how to apply PLCS (see Section 3.3). Yet another meta-model can be a guide for how to use DEXs for a specific domain, e.g. by means of a concept model (see Chapter 6). Another level above could specify a guide for how domain-specific guides should be created. In general, everything can be administered, and the administration can be administered as well.

There are also other information dimensions that adds complexity, e.g. the functionalities provided by Product Data Management (PDM) systems. Such functionalities include versioning of products and documents, owners of information (persons and organisations), roles, views, and effectivity information (validity). Additional complexity comes with traceability of decisions and requirements, change management, etc. Moreover, other viewpoints found in Zachman’s ISA framework.
Figure 8.2: Differences in information context due to hierarchical levels such as abstraction levels, detailing levels, and meta models

(see Section 3.4) adds even more dimensions by not only addressing the question what is to be modelled, but also how, where, who, when and why.

8.2 Is one common information model realisable?

From the multi-dimensional complexity exemplified above it is evident that a model that aims to satisfy the many dimensions of the manufacturing system life cycle will inevitably be complex as well. Moreover, in reality there are seldom discrete borders between two levels or phases but rather continuous and fuzzy ones. Each dimension seems manageable in isolation, but how will the resulting model be if all dimensions are combined? This section will reason about whether it is realistic or not to represent manufacturing system life cycle information in one common model.

Inevitable delimitations

To start with, it is obvious that an ambition of representing “everything” in a model is unrealistic, since a model always has a border to a greater context as well as a limit regarding its detailing level. Even if the manufacturing system life cycle scope is large, it is still a part of a wider context, e.g. enterprise frameworks such as
8.2. IS ONE COMMON INFORMATION MODEL REALISABLE?

Figure 8.3: If the model is too complex, will its usefulness decrease?

Zachman and CIMOSA (see Sections 3.4 and 3.3). Since the world is continuous, an appropriate delimitation of the scope is inevitable.

Regarding detailing level, the area of manufacturing system development and operation includes many sub-areas, such as product design, process planning, investment process, flow simulation, operation and maintenance, etc. Within each sub-area, a new world of details opens up. A model can be very detailed, but there is always a need for simplification, or the model would be exactly like the reality. An appropriate detailing level is thereby also inevitable.

Model complexity – opportunities vs. usability

It makes sense that a larger scope and higher detailing level makes a model usable for more applications, which in turn provides an information base for more advanced reasoning. Although an information model with a large scope and many details is useful for many purposes, such a model would likely be difficult to build and maintain. A simple model, on the contrary, is easier to build and maintain, but is of course applicable for fewer applications.

Is there a limit of how complex a model should be? It is assumed that there is. At some point, the usefulness might decrease due to the model’s complexity as illustrated in Figure 8.3.

To motivate such an assumption, an analogy can be drawn between the choice of model complexity and the choice of an appropriate number of boxes to store one’s belongings. The choice of having one box for all things implies no order at all and consequently a long search time. An analogous model would have only one universal information type. Two or three boxes would increase the order since there are now categories. The other extreme is to have one box for each object. Again, there would be no order since each object is its own category. Such an information model does not exist, since it would be exactly as the reality. Between these extremes, the achieved order is first increasing, up to a point where the number of boxes becomes too hard to take in. Past that point, the usability decreases, since one can
no longer remember where things were put. The adequate complexity of a model should thereby be before that point, at least from a practical viewpoint, since there is a limit for how much complexity we can take in.

**The manufacturing system life cycle scope – too complex for one model?**

Where the limit of an appropriate model complexity is, has not been evaluated. However, it is considered likely that the information scope of this work, i.e. information for developing and operating a manufacturing system, becomes too complex to define in one single model, if all different levels and time aspects described in Section 8.1 are to be included.

The reasoning starts with the many applications that are used during the manufacturing system life cycle. Taking a particular resource as an example, information about this resource occurs in various computer applications, as shown in Figure 8.4. In each application, certain aspects of the machine are captured in a format that is adapted to the application’s purpose and internal data model. This means that resource models can vary in both format, information structure and terminology. Moreover, the instantiated data can vary in terms of abstraction level, life cycle stage, decision stage, and other dimensions of complexity.

The idea of defining a neutral model for the common information is theoretically appealing. Two variants can be imagined, as illustrated in Figure 8.5 and Figure 8.6.
8.2. IS ONE COMMON INFORMATION MODEL REALISABLE?

The information models of various computer applications are drawn as cylinders, where the scope is represented by the wideness of the cylinders, and the level of “specificness” is represented by their heights. A specific model uses more specific information concept than a generic model (e.g. “Machine tool” is more specific than “Resource”).

Figure 8.5 shows one variant where all common information is modelled according to a “universal” standard model. Such a model would theoretically eliminate the need for data mappings since all information can be described in the same model. The universal model is drawn as a cylinder that encloses all models needed in various systems and applications during the manufacturing system life cycle. Such an information model is however generally considered utopian (Wijnker, 2003). As Wijnker (2003) describes, an information model requires a consensus on the formalisation of concepts, which is realistic only within limited contexts. For this reason, it is probable that several standards and ontologies will coexist within the manufacturing domain also in the future.

Moreover, an imaginary universal standard covering the whole manufacturing domain could lose the initial idea of using models. A model is supposed to capture aspects of the world in order to answer just the questions that are of interest. Too many interesting questions and aspects makes the model too much like the complex reality the model was supposed to simplify. As was assumed earlier, the complexity of such a model could also reduce its usability (see Figure 8.3).

It is however not stated that the manufacturing scope is too large to model – it all depends on how specific the model is. For this reason, standards capturing a large scope are usually, probably always, designed to be generic. In Figure 8.6, a generic integration model is drawn like a bottom plate that is wide enough to be able to integrate all applications, but at the same time flat (generic) in order not to over-define and be in conflict with the models of all applications. It functions as the least common denominator for the specific applications. Models with such characteristics already exist, e.g. the STEP standard AP239 (PLCS).

**Strictness vs. freedom**

The applicability of PLCS for covering the scope of manufacturing system life cycle was acknowledged in Chapter 5. The benefit of a generic information model is its wide applicability and versatility and its resistance to changed circumstances. However, as was described in Chapter 6, a higher degree of freedom and versatility leads to instantiation uncertainties, such as which entity to use. A specific single-purpose model is more secure against inconsistent instantiation since it forces instantiation to a strict format.

Steps towards reducing inconsistency problems have been initiated with the development of standardised reference data libraries (RDL) and data exchange sets (DEX) (see Section 3.3). Reference data is used to classify information entities of the standard model into more specific information concepts, and DEXs and Capabilities guide how to use the modules of the PLCS standard. However, since
RDL and DEXs are domain-independent, they do not provide support in domain-specific applications, such as when one would like to instantiate a machining centre. For this reason, a concept model as presented in Chapter 6 was developed. A concept model represents knowledge regarding what information to model and which terms to use. Moreover, the purpose was also to guide how to apply PLCS. It was however found that a simple relation such as “has” between two concepts in the concept model can be mapped to various relationship entities in PLCS, depending on how the associated concepts are represented. This has previously been shown in Figure 6.5. The inconsistency problem remains due the versatile and generic character of PLCS as well as the large gap between specific concepts of the domain and the generic concepts of the PLCS standard. This gap is illustrated by the space between the specific models and the bottom level in Figures 8.5 and 8.6.

Two approaches to cope with inconsistencies

In order to cope with the inconsistency problems described above, two possible approaches can been imagined:

1. Building strict instantiating templates
2. Allowing inconsistent instantiation (and find a way to deal with it)

The first option indicates that there is a need to agree on only one way of instantiating e.g. a machining centre. This implies that for each concept, one or many instantiation rules are necessary. This would be a more complicated alternative to a scenario where a standard specifically aimed for representing machine tools had
been used. If the particular integration scenario does not exceed the scope of the machine tool standard, the use of a generic standard does not have a clear purpose.

The second option is to allow different variants of instantiation. Since the number of instantiation variations greatly outnumbers the number of computer systems, the idea to use standards to reduce the number of translations fails if the translation is based on a fixed information structure. For this reason, instead of relying on a fixed information structure, an alternative way to find information is needed. The use of concept models as described in Chapter 6 can function as a basis for information search by utilising the ability to classify PLCS entities according to the common terminology. A term-based search works well for humans, but from a system integration perspective, it is likely to be more complicated compared to integration based on fixed information structures.

Since the generic model of Figure 8.6 requires complementary domain models and an extensive instantiation guidance, there is a practical factor concerning whether the integration benefits of a common generic model is worth the cost of developing and administrating its guidance. Thus, the question whether one common information model is realisable or not, is not answerable without introducing a cost factor. When time and resources are limited, as they are in real life, the question should rather be: How much can an information model include in order to give sufficient integration benefits to be worth its costs?

To conclude, it is obvious that within the scope of the manufacturing system life cycle, there is no easy way to represent all information according to one model. A generic model can be used for integrating a lot of information, but due to the instantiation guidance issue, it is not considered practical to rely on such a principle solely. A certain degree of heterogeneity is unavoidable. Therefore, a more pragmatic solution is to apply specific models where such models suffice in terms of integration. As will be exemplified later, information can be integrated in other ways than sharing a common model.

### 8.3 Choosing a model architecture

The reasoning above leads to the conclusion that there is a need for more than one model, and that the initial hypothesis (“In order to integrate information from the development phase and the operation phase, information types from both life cycle phases need to be represented in one information model”) needs to be revised. As an alternative to the initial hypothesis, the suggested approach is to find suitable delimitations of models and to make them complement each other. A model architecture with complementary models is suggested for the manufacturing system life cycle.

#### A mix of abstraction levels

First, it is obvious that there is not only one generic level and one specific level, but various intermediate levels as well. Figure 8.7 illustrates that information concepts
exist in various abstraction levels and scopes. At the bottom of this pyramid, the most specific concepts are found, such as machining centre, rough milling, spindle speed, cycle time, and tool length. Such concepts are used in various computer applications for engineering activities such as process planning. Also the machine tool standard ASME B5-59.2 includes concepts of this level.

In order to integrate specific domain concepts, more generic domain concepts can be used. For example, AP214 includes generic concepts such as process operation, mass property, and duration property. Such concepts are sufficiently generic to be applicable within its scope and sufficiently specific in order to ensure some degree of consistency.

In order to relate generic domain concepts, more generic concepts are needed, such as part, activity and property. Those types of information concepts are found in PLCS.

Still more abstract concepts are usually found in ontologies where top down categorisations starts from the universal object “Thing” (e.g. ISO 15926). Such abstract concepts are likely to be too unspecific to be useful for manufacturing system modelling.

In reality, there are of course no sharp borders as pictured in Figure 8.7. On the contrary, most information models and standards include concepts from many different abstraction levels.

However, it is likely that the use of an unnecessarily generic model for handling
very specific concepts will result in an unnecessarily complex information management due to the additional needs of instantiation guidance. Moreover, the more generic a standard is, the more versatile and flexible is it likely to be. For this reason generic models are usually the most circumstantial, since they need to take several viewpoints into consideration. This motivates the use of more specific models when such models suffice.

For this reason, the probable integration scenario within the manufacturing system life cycle would not be as pictured in Figure 3.5, but rather more heterogeneous as pictured in Figure 8.8. Hybrid approaches have also been suggested by Wijnker (2003), since different ways of integration have different pros and cons in terms of e.g. cost and time. Integration according to specific standards, or occasional point-to-point mappings where such mappings suffice, should therefore be justified from a pragmatic perspective.

A separate runtime data model

Regarding life cycle aspects, many factors points to the need for a separate runtime data model. In Section 7.4, it was argued that a generic model such as PLCS has too much overhead to make it optimal for representing momentary data.

In addition, the development phase has high requirements on modelling flexibility since it consists of intertwined sub-processes where information needs to be communicated and shared while it is created. In the operation phase, the scenario is different since information can be gathered and created for later use. Thus, the runtime data model can thereby be designed as a data asset.

A separation of information models does not exclude the possibility to integrate information from these phases, since integration can be achieved by other means, such as implementing synchronisation routines between different models. For example, as-is instances of resources can be updated with runtime measures originating from a separate runtime data model. In this way, runtime measures can be related
to an as-is instance of a resource model without having to define the origin and context of this measure. To get this context, the attached runtime data should either be synthesized into a form where the necessary context is given, or the context has to be analysed separately.

**Knowledge representations and application logic**

Another aspect is to what extent information models need to be designed to ensure a consistent instantiation in relation to complementary mechanisms that can control instantiations, such as meta-models, knowledge representations and application logic.

For example, an information model sometimes needs to control dependencies between attribute values. In the example shown in Figure 8.9, the value of the attribute `name` of `Operation_type` is dependent on the attribute `name` of `Tool_type`. It would be strange if the machining centre was modelled to perform a milling operation using a drill as tool. Since such dependencies require knowledge of the domain, complementary rules are needed:

\[
\text{tool\_type(drill)} \rightarrow \text{operation\_type(drilling)}
\]

\[
\text{tool\_type(mill)} \rightarrow \text{operation\_type(milling)}
\]

Such logical checks can also be implemented in the application logic. Applications should thereby not only solve engineering problems but also ensure a consistent information creation.

The presence of knowledge representations and application logic also introduces more choices regarding the interaction between different models. Application logic can conjure relations and dependencies between models as well, and thereby function as a glue between information models, such as between a runtime data model and a resource model.

This exemplifies that information modelling is a piece of a larger information management context that has the overall purpose to provide a person with information necessary for making good decisions.
Concept models and terminology

Since a heterogeneity of models usually reduces integration, there is a need for an ingredient that brings forth community. One such ingredient is a consistent use of terms. After all, it is not the exact structure of information that is the most important, but the meaning of information concepts, and the terms used for symbolising them. Terms have a key role when searching for information outside one’s reach.

Thus, the development and utilisation of concept models is considered important, regardless of the choice of models. As has been stated in e.g. Chapter 6, the use of generic standards such as PLCS makes it important to classify information concepts according to more specific reference data classes based on the terms of the concept model. This is to know what the generic concepts actually represent. Information concepts of more specific models, as well as implementations, should also be symbolised with terms according to the concept model.

To enable such classifications of information concepts, a concept model needs to be represented in a computer-interpretable format such as OWL. Moreover, in order to facilitate accessibility and readability from a human perspective, concepts models should be implemented as navigable concept browsers similar to the Conzilla browser presented in (Naeve, 2001).

An alternative modelling scenario

As a result of the discussion and reasoning above, an alternative modelling approach is summarised in Figure 8.10. The numbers are explained in the list below.

1. Standardised models are useful for communicating information between various systems during the manufacturing system life cycle, e.g. resource models, process models, geometric models, etc. Since the activities of the development phase need to share a lot of common information, the communicated information should be integrated according to a neutral model or standard. An ideal model to use in order to minimise instantiation inconsistencies, is one that in terms of abstraction level and scope lies “closest” to the set of information that needs to be integrated.

2. In cases where specific information standards need to be integrated into a larger context, a more generic standard can be used, such as AP239 (PLCS). Two variants are possible: by explicit mapping from a specific standard to PLCS (according to guidance), or by attaching embedded models.

3. In order to analyse trends, runtime data need to be collected periodically according to a separate runtime data model aimed for providing a data asset. Such a model can be the basis for a runtime database.
4. In order to feed back operation experience, runtime data can be synthesised into more interpretable formats, such as diagrams and measures. Such feedback can be associated to the as-is representations of real resources.

5. As a complement to analyses on a line level or factory level, local information management should be possible for local issues, e.g. within a machine. Local models could also be used to ask resource individuals directly about their status.

6. Integration of information from both life cycle phases is approached by common ontologies and concept models of the domain, indicating a preferred terminology. A consistent use of terms for all models is important to be able to find information residing in other models.

In order to exemplify the use of embedded models as an alternative to explicit model mapping, the following scenario is depicted in Figure 8.11: An instance of the generic model AP239 (PLCS) has instances of three “products” (machine tools) in the context of three different life cycle stages, using Part (as-designed), a Product_as_individual (as-planned), and a Product_as_individual (as-is). In order to represent these “products” in detail, attached XML files define machine tool instances according to a machine tool standard, such as ASME B5-59.2.
8.4. DISCUSSION SUMMARY

Since ASME B5-59.2 does not include all details of a machine tool, complementary models are referred to. The as-designed instance refers to a geometric model defined according to AP203, and the as-is instance refers to an elaborated accuracy analysis.

In this approach, a PLCS instance has the functionality of relating other instances in a larger context without the need to explicitly represent each instance. Such an approach has however drawbacks similar to document-based information management, since information in embedded models is more difficult to access. As usual, a benefit from one perspective leads to a drawback from another. This, however, exemplifies another way of integrating information than explicitly defining information according to one information model.

The presented model architecture and instance examples are merely ideas resulting from the discussion in this chapter. An elaboration as well as an evaluation of alternative integration scenarios are left for future work.

8.4 Discussion summary

- There is a difficult balance between standardisation and freedom.
- Since it is not realistic to represent information from all viewpoints and complexity dimensions of the manufacturing system life cycle according to one common information model, alternative ways of integration needs to be considered. A model architecture with several collaborating models is needed.
• When possible, specific models should be integrated by using a standard of the closest possible abstraction level in order to ensure a consistent instantiation.

• The importance of a common terminology is evident. Information is likely to be found in one way or another, if one knows what terms to search for. The development and implementation of concept models is proposed, e.g. in OWL format.

• The practical aspect of modelling need not be forgotten. Theoretically superior approaches need not be the most practical and value-adding in real life.

• Information can be integrated in more ways than sharing a common model, e.g. via knowledge models, application logic, and embedded models or documents. From a practical viewpoint, there should be an optimal mix of these ways that enables information access to a decision-maker with a minimum of effort.
Chapter 9

Conclusions

In this chapter, conclusions are presented for the research questions, the hypotheses, and the discussion points. Finally, the need for future work is presented.

9.1 Conclusions regarding research questions

RQ1

How should resource information be modelled to be utilised for different purposes throughout the manufacturing system life cycle?

- Initially, an information standard that covers the scope of the manufacturing system life cycle was identified. Hypotheses regarding the use of the STEP standards AP214 and AP239 were formulated and tested using data from case studies:

  - The sufficiency of AP214 for the life cycle perspective was falsified, due to limitations in representation of different life cycle phases of the resource. In particular, the planning perspective of a resource is limited. Moreover, since AP214 has a development focus, it lacks explicit representation of information relevant for the use/operation phase.

  - A mix of AP214 and AP239 is promising in terms of information content and combined scope, but the approach is rejected due to consistency issues and potential violations of the standards.

  - The sufficiency to use AP239 together with external reference data libraries (RDL) is promising in terms of scope. Thus, AP239 has been chosen as the most suitable STEP standard for representing manufacturing resources from a life cycle perspective. Suitable features of this standard are e.g. representation of different life cycle stages (as-designed, as-planned, as-realised/is) and the ability to form several breakdown structures (e.g. zone, functional, and system breakdowns). However,
even if the sufficiency of AP239 can be theoretically corroborated, there are issues from a practical point of view. The use of such a generic standard needs further instantiation guidance in order to ensure consistency and the ability to retrieve information. Such guidance needs to be more domain-specific than the current DEXs, e.g., how to represent a machining centre.

• As a response to the need for specific instantiation guidance, an initial concept model of a machining centre has been presented, showing which aspects/concepts of a machining centre are important from a user perspective. A concept model can serve several functions:
  – It is a kind of knowledge representation of the domain
  – It provides a common terminology where the terms can be used for information search in various models
  – It defines what information is important to represent (from a certain perspective)
  – It can be used as a guidance for how to instantiate generic models

• However, the flexible and generic character of the PLCS model results in numerous ways to use PLCS for representing specific machining centre concepts. This means that the consistency problem remains. From a practical point of view, the use of more specific standards can be motivated, where such standards suffice. More about this is described in the conclusions of Section 9.2.

Regarding the last part of the research question “throughout the manufacturing system life cycle”, runtime aspects are elaborated below in the conclusions for research questions 2 and 3.

RQ2

*How can runtime data be used to ensure that there is an accurate model of the manufacturing system?*

• An as-is instance of a resource is a specialisation of an as-planned instance, with the addition that it can hold values about runtime data as well. Due to the dynamic character of the runtime data, it is not suggested that such models should capture all runtime data, since the updating requirements would be tremendous. Instead, the as-is instances should include properties and analysis results that are of interest during manufacturing system reconfiguration. The following has been concluded:
– Runtime data that can be excluded from an as-is representation are momentary data, since the validity of such information is perishable. Examples of snapshot data are current status of the resource (e.g. up or down) and snapshots of the process (e.g. latest cycle time). Such information should rather be retrieved from the source, i.e. the equipments directly, or a database that can log such information.

– What could be included in a as-is instance of a resource are activities such as maintenance and the latest values of capability properties and process-related measures such as MTBF, MTTR, cycle time, etc. This would rely on updates from systems that calculate such values.

– Finally, what should be represented in an as-is instance of a resource is information that could be relevant for manufacturing system reconfiguration, such as statistics regarding failures, accuracy, and performance measures. Since the receiver of such information is likely to be a person, there is a possibility to represent such information as documents associated to the as-is instances.

RQ3

*How can experience from the operation phase be utilised in the work processes of all life cycle phases?*

- Runtime data are likely to be used if the use is explicitly defined in the work processes of development activities. For example, during reconfiguration of a manufacturing system, tasks such as “check the recent accuracy evaluations of resource individuals” should be included.

- Runtime data need to be interpreted within the context where the data were created, especially when it is to be fed back to development. It should be possible to analyse resource behaviour with respect to many aspects, such as used tool, used feed rate, etc. The manufacturing context has been defined as the process context supplemented with a runtime context. The process context includes the process (e.g. process type, tool type), the used resource and its configuration, and the produced product (e.g. feature, material). The runtime context contains a snapshot of other runtime data from the same measurement time, including the environment data such as temperatures and humidity. Reliable analyses relies on periodic logging of parameters and automatic event reporting.

- Runtime data (as defined within this work) can range from raw data to experiences and cases. A more synthesised form of runtime data is likely to be more interpretable for humans, especially when used in development activities. Raw data is necessary for analyses, but in order to communicate results
of an analysis, a synthesised format such as mean values, diagrams, reports, etc, increases interpretability.

- As stated earlier, it is not convenient to represent momentary runtime data in PLCS format due to its voluminous format lack of explicit runtime data types. Moreover, the purposes and characteristics of runtime data models are quite different compared to development models (see Section 8.1 and 8.3). For this reason, a separate runtime data model is suggested for the purpose of gathering runtime data. Based on this data, analyses can be made and results can be associated to as-is instances of resources to be used when the capability of resource individuals needs to be evaluated.

- The operation phase also offers many scenarios for learning and intelligent reasoning on the local level, i.e. for each machine individual. An example based on machine alarms has been described. The trend towards more intelligent products can possibly reduce the need for information management if an as-is representation of a resource could be updated by use of sensors and control systems.

### 9.2 Conclusions on research goal and initial hypothesis

The research goal was to find a modelling approach that would contribute to the accessibility and integration of manufacturing system information within and between the development and operation phases of the manufacturing system life cycle, and come closer to the desired information management characteristics, which were:

- **Accessible**: Information should be easily found, without wondering where to search and how
- **Reusable**: Information should be easily fetched and refined in other applications and contexts
- **Communicable**: Information should be easily exchangeable between various computer applications
- **Interpretable**: The meaning of information should be interpretable for both humans and computers
- **Well-structured**: Information should have its logical place without any ambiguity about where to put it or in what format

From the discussion in Chapter 8, it can be concluded that it is difficult to achieve all these characteristics by the use of one model only. The interpretability and well-structuredness can be achieved by using specific models. Communicability can be achieved by using generic standard models. Accessibility can be achieved by integrated models, or by a consistent use of terms, since terms can be used
for information search. *Reusability* is achieved if all the other goals are achieved. Much of the responsibility regarding accessibility and reusability lies of course on the implemented functionality of computer systems such as providing application programming interfaces and import/export functionalities.

Information defined in one information model can be more easily used for comparisons and reasoning. However, to define *all* relevant information aspects of the manufacturing system life cycle according to one single model has been considered too complex to be useful. It is assumed that there is a point where the complexity of models leads to more maintenance costs than integration benefits. It is further believed that the complexity of the manufacturing system life cycle is past that point.

Since a generic model has the problem of ensuring consistent instantiation, the concluded approach is to find appropriate delimitations and interfaces between complementary models with different purposes, scopes and abstraction levels. A model architecture has been exemplified in Figure 8.10 in Chapter 8. In order to compensate for the reduced integration effect implied by the use of many models, information concepts in different models need to be classified according to a common terminology.

Further, the desired information management characteristics indicate the objective of aiding the activities of the manufacturing system life cycle. The administration regarding how to use a model should not outgrow the model’s intended benefits. Moreover, information models do not have the sole responsibility of integration, since ontologies, data models, the functionality in application logic, etc, are other important factors.

The human involvement should also be remembered. Whereas computer systems need logical relationships between information types, humans can understand such relationships by experience. The practical aspect of information modelling and information management should thereby not be excluded. In addition, the choices related to information modelling need to be adapted to the needs of each particular situation.

Consequently, since one information model for the manufacturing system life cycle is not realistic from a practical viewpoint, and since there are complementary ways of achieving information integration, the initial hypothesis “*In order to integrate information from the development phase and the operation phase, information types from both life cycle phases need to be represented in one information model*” needs to be falsified as it is formulated.

A possible hypothesis for the future is instead:

*In order to integrate information from the development phase and the operation phase, information types from both life cycle phases need to be represented using a mix of specific and generic models in order to achieve both well-structuredness and integration. Integration is facilitated by classifying information concepts according to a common concept model.*
The verification of the presented model architecture and the revised hypothesis is saved for future work.

9.3 Future work

Since the scope of this work is wide, many areas have not been surveyed in detail and need to be further elaborated. Some examples are:

- Continuation of the development of concept models of a machining centre as well as other resources and their interfaces to neighbouring concepts such as factory, layout, process liquids and tools.

- The possibilities of representing concept models in OWL format to be used as reference data for various standards needs to be elaborated. In addition, the implementation of concept models as navigable concept browsers should be considered.

- Further elaboration and testing of the suggested model architecture.

- Further evaluation of upcoming standards related to machine tools, such as STEP-NC Part 110 and ASME B5.59-2. Moreover, the possibilities of mapping such standards into PLCS need to be evaluated.

- Further assessment of the DEXs for PLCS.

- Further investigation of the possibilities of applying decentralised information management, such as agent programming, for the operation phase.
Bibliography


Astrakan Strategisk Utbildning AB, (2003) Begreppsanalys med Modellering, Course Material


Lipton, P. (1991), Inference to the Best Explanation, New York:Routledge, 1991 (Excerpt of (Curd and Cover, 1998)).


ModArt (2007), ModArt – Model Driven Parts Manufacturing, The MERA program Projects 2007, ISSN 1650-3120, p170-176


NE – Nationalencyklopedin online (2006), Search term: klass, http://www.ne.se/

NE – Nationalencyklopedin online (2006), Search term: ontologi, http://www.ne.se


OASIS (2008), online documentation available at http://www.oasis-open.org/


PDTnet project documentation online (2004), available at

PDTnet XML Schema (2003), available at


Rote buch online (2008), available at http://www.rotebuch.de/index2.html


