TAMPERE UNIVERSITY OF TECHNOLOGY
Degree Programme in Information and Knowledge Management

JANNE KIIRIKKI

UBIQUITOUS MANUFACTURING REQUIREMENTS FOR PRODUCT DATA MANAGEMENT

Master of Science Thesis

Examiner: prof. Samuli Pekkola
Examiner and the topic have been approved at the Council Meeting of the Faculty of Business and Technology Management on December 8th 2010

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master’s Degree Programme in Information and Knowledge Management
KIIRIKKI, JANNE: Ubiquitous Manufacturing Requirements for Product Data Management
Master of Science Thesis, 72 pages, 6 Appendix pages
January 2011
Major: Information and Knowledge Management
Examiner: Professor Samuli Pekkola
Keywords: Ubiquitous manufacturing, ubiquitous computing, pervasive computing, product data management, PDM, digital product process, product structure

Ubiquitous manufacturing rather straightforwardly means the usage and utility of ubiquitous computing concepts and technologies in manufacturing. Simply put this means embedding more intelligence into objects of the production line, making them communicate and using all data this generates to further enhance the functionality and visibility of the production.

The study is structured into theory and empirical sections. Theory part is a literature review for defining the concepts of ubiquitous computing, product data management and ubiquitous manufacturing. Along with the definition of ubiquitous manufacturing, ubiquitous computing ideas are used to build ubiquitous manufacturing concepts. These definitions and concepts are then used as a foundation for defining the requirements that the ubiquitous manufacturing sets on product data management. The requirement definition mainly studies the role and availability of product information, regarding different phases and aspects of the product and its manufacturing.

In the empirical part of the study a data model is built for product structure management, which aims to fulfill the needs and requirements posed in the theory section of the study. The developed data model is also put into test in a case study section, where a pilot production line, that realizes single ubiquitous manufacturing concept, is built at The Technical Research Centre of Finland VTT. The conclusion of the study discusses the implications of the study and of ubiquitous manufacturing in general.

The study concludes that one of the key possibilities of ubiquitous computing in manufacturing environment is the development of product data management to the level of real-world single item management. This real-time information of singular items enables development of production through better controllability, monitoring and optimization. In a ubiquitous manufacturing concept such as this, the main requirements for PDM are real-time singular level information management, and possibility to model product manufacturing phases in addition to modeling products.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Tietojohtamisen koulutusohjelma
KIIRIKKI, JANNE: Ubiikin valmistuksen vaatimukset tuotetiedonhallinnalle
Diplomityö, 72 sivua, 6 liitesivua
Tammikuu 2011
Pääaine: Tiedonhallinta
Tarkastaja: professori Samuli Pekkola
Avainsanat: Ubiikki valmistus, jokapaikan tietotekniikka, ubiikki tietotekniikka, tuotetiedon hallinta, tietomalli, tuoterakenne

Ubiikilla valmistuksella tarkoitetaan varsin suoraviivaisesti jokapaikan tietotekniikan käyttöä valmistuksen apuna. Yksinkertaisesti tämä tarkoittaa tiedon ja älyn lisäämistä tuotantolinjan kappaleisiin, kuten työkaluihin ja komponentteihin, näiden kappaleiden välistä kommunikointia sekä tästä syntyvän uuden tiedon käyttämistä toiminnan analysoimiseen ja kehittämiseen.

Tämän työn pääasiallinen sisältö on tutkia millaisia vaatimuksia ubiikki valmistus aiheuttaa tuotetiedonhallinnalle. Työ jakautuu teoria- ja empiriaosuuksiin. Teoriaosuudessa aihetta lähestytään kirjallisuusselvityksen keinoin ubiikin tietotekniikan ja tuotetiedonhallinnan määrittelyn ja vaatimusten kautta, siirtyen lopulta ubiikkiin valmistukseen. Samalla tutkitaan lyhyesti ubiikin tietotekniikan mahdollistamia konsepteja ja uusia toimintatapoja valmistusympäristöissä. Näiden konseptien pohjalta tehdään vaatimusmäärittely siitä, mitä vaatimuksia ubiikki valmistus aiheuttaa tiedonhallinnalle ja etenkin tuotetiedonhallinnalle.

Empiriaosuudessa muotoillaan tuotteelle tietorakenne, joka vastaa ubiikin valmistuksen tarpeisiin ja vaatimuksiin. Tätä tuotteen tietorakennetta testataan vielä lyhyesti case-tutkimuksena toteuttamalla Teknologian Tutkimuskeskus VTT:n Espoon tiloissa kokeellinen tuotantolinja, johon on liitetty ubiikkia toiminnallisuutta. Lopuksi työssä käsitellään vielä tutkimuksen tuloksia sekä ubiikin tuotannon vaatimuksia ja mahdollisuuksia yleensä ottaen.

Keskeisinä ubiikin valmistuksen mahdollisuuksina työssä nousee esille tuotetiedonhallinnan kehittäminen jokapaikan teknologioiden avulla reaalimaailman kappalekohtaiselle tasolle, jossa on keskeistä erottaa, mitä tietoa kustakin yksittäisestä tuotteesta talletetaan millekin tasolle. Tämä reaaliaikainen kappalekohtainen tiedonhallinta mahdollistaa tuotannon tehostamisen muun muassa paremman ohjattavuuden, optimoinnin sekä tarkkailun kautta. Tällöin tuotetiedonhallinnalle asetettuja päävaatimuksia ovatkin juuri reaaliaikainen kappalekohtainen toiminta sekä valmius mallintaa itse tuotteiden lisäksi myös tuotteiden valmistusvaiheita

PREFACE

This master's thesis was written at VTT for the UManu project exploring the possibilities of ubiquitous manufacturing for Finnish manufacturing industries. The VTT project worked as a great framework and formed the topic for this master's thesis. I also found the topic very interesting, and time to time felt the joy of studying something of novelty, something that possibly not too many thesis workers already have scoured upon.

The thesis itself formed without any larger problems and on schedule. One of the biggest reasons for this being the invaluable possibility to be able to throw monumental amounts of paid working hours into it, a desirable luxury most of us thesis workers do not get to enjoy.

My thanks go to the examiner of this thesis professor Samuli Pekkola, whose comments kept the whole package on tracks and generally pushed things forward, and to my supervisor at VTT, senior research scientist Mikael Haag, whose project and ideas the thesis is based upon, for his major help in cultivating my knowledge of the subject.

In addition, I owe my deepest gratitude to my partner Janni, who whole-heartedly supported me during the entire process, even through some rather unthankful times in our lives.

Helsinki, January 17th, 2011

Janne Kiirikki

TABLE OF CONTENTS

ABBREVIATIONS, ACRONYMS AND NOTATION

API	Application Programming Interface
BOM	Bill of materials, list of subcomponents that build up the component of product
CIM	Computer Integrated Manufacturing
Configuration	Arrangement of objects
Customization	Differentiation or personalisation of objects
Database	Information system for organized collection of data
ERP	Enterprise Resource Planning system
Flag	Single identifier depicting some quality or option
Fig.	Figure
GPS	Geographic Positioning System
IS	Information System
Mass Customization	Functionality of customizing items with various options en masse
MES	Management Execution System
Ontology	Formal, explicit specification of a concept or system, containing all related objects and their relations.
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
PDM	Product Data Management
PIM	Product Information Management
PLM	Product Lifecycle Management, concept for managing product data throughout its life all the way from designing and manufacturing to recycling and disposal.
Product	Term for all the products of certain single type.
Process	Particular course of action intended to achieve a result
Revision	Act of revising, altering or reorganizing something
RFID	Radio Frequency Identification
Single product	Term for a single manufactured unit of product
SME	Small and Medium Enterprises
Topology	Anatomy of a structure
UHF	Ultra High Frequency, term used for electro magnetic wave frequencies ranging from 0.3 GHz to 3 GHz.
UML	Unified Modelling Language, modelling method used in this document
UWB	Ultra Wide Band, positioning technique that uses wide band radio waves and multiple antennae on determining the source of the waves
Version	Snapshot of a concept or system at a certain point in time
XML	Extensive Mark-up Language

1. INTRODUCTION

In an ideal factory of the future, all the pieces, tools, objects and people know where to be and when. Helped by this information, the burden of workers is greatly eased, as they can only follow the instructions generated by the process and supervise that everything is in order. The management system of the factory knows the exact location of every single nut, bolt and tool, making it easy to measure and follow the production. New orders can be customized according to customer’s wishes, and dynamically allowed to production, where orders can be timed and calculated according the delivery dates automatically. When a product leaves the factory, all information about its manufacturing steps and components can still be retained for later reference. It is a futuristic vision with seamless information flow indeed, but one that we are possibly getting closer to day by day.

Modern computing has had an impact on everyday lives of people through the usage of communication technologies and internet. Nevertheless industrial level solutions in manufacturing seem to lag behind the state-of-the-art computing. It is even argued that computer-integrated manufacturing (CIM) has not been facilitated successfully in large scale even though it has been around since 1980’s (Zuehlke 2010, pp. 129-130). Even as the idea of computer-integrated manufacturing has been around, the lack of standardization and technologies have kept it at bay (Meyer et al. 2006, pp. 6-7). In regard to the unceasing progress of technology, modern computing technologies and techniques still hold quite a potential for manufacturing to exploit.

The scope of this thesis is to study the field of modern manufacturing technologies from the information management point of view. Ubiquitous computing can be seen as a possible paradigm shift in the future of computing. This thesis aims to clarify its possibilities and requirements for usage of ubiquitous computing in manufacturing environment, especially from the product’s point of view. The product data management (PDM) is taken as a foundation for enhancing the product related data through the whole manufacturing.

Another closely related area of interest is integration of product development, product data and assembly line definition. Lanz (2010) discusses about the unified ontology through all the phases of manufacturing, which can be seen as a possible functional framework for ubiquitous manufacturing as well.

Current motivation for product data management (PDM) development comes mainly from increasing the complexity of the products and the requirements for faster product development (Kovacz et al. 1998, p. 285, Kropsu-Vehkaperä et al. 2009, p. 759). Product information digitalization is also currently seen as rising trend (Westkämper & Jendoubi 2003, p. 1), as digitalized information is one of the key requirements in order to manage product lifecycle (PLM). The interest towards PLM has risen most of all through the need for more seamless integration of product data throughout all the phases of its life (Thomas et al. 1999, p. 54). It makes no sense to manually convert all data to fit into various stages of product life.

To further engrave the role of knowledge management in manufacturing, Westkämper & Jendoubi (2001, p. 1) define information about manufacturing resources being a critical element for manufacturing companies. Kropsu-Vehkaperä et al. (2009, p. 758-759) also define that efficient data management practices are one of the key aspects for business efficiency.

1.1. Scope and objective of the study

The primary research question this study is aiming to answer is

- What requirements does ubiquitous manufacturing set for product data management?

Other research questions include such as

- How to formulate PDM that fulfils these requirements?
- What data does ubiquitous manufacturing need? What data does it provide?
- How to implement and fulfil these requirements in manufacturing information systems?

The goal of the thesis is to construct a high level model (concept) for an information system in order to support PDM in manufacturing environments where ubiquitous technologies are present. The developed model is also empirically evaluated, as a part of the study also contains a case study of developing and realizing a single ubiquitous manufacturing concept. The ubiquitous manufacturing concepts that are developed during the study are also used to pose the requirements for PDM. Also the results of the VTT Technical Research Centre of Finland’s ubiquitous manufacturing project are heavily linked into the empirical section, as the case study part of the thesis is mainly a report of the project realization.

The main limitation for the study is to limit the product data into the data used in manufacturing phase (beginning of life, BOL). Therefore product lifecycle management for later parts of product’s life is not given any regard. As the study is focusing on manufacturing, the ubiquitous aspect of the final product received by the customer is not included in the study. Also manufacturing is mainly seen from a Finnish SME manufacturer's point of view, which means that the emphasis is on computer aided manual manufacturing, instead of fully automated manufacturing.

1.2. Background

The manufacturing environment can be thought to consist of three different parts: processes, products and people. Products and processes are generally "unintelligible" in the regular manufacturing context, whereas people can reason and communicate more or less seamlessly. The ubiquitous manufacturing concept aims to increase the presence of this intelligence from people to include products and processes as well. The increment of intelligence can manifest for example as products and processes that can communicate with each other in order to automate and enhance manufacturing system further.

In order to manage all the new information that is related to this ubiquitous concept, data and knowledge management in the manufacturing concept has to be cultivated. Product data management (PDM) system can be seen as a base for developing manufacturing data management in order to support ubiquitous manufacturing concepts.

Some authors (Meyer et al. 2009, pp. 6-8, Lucke et al. 2008, p. 115) also closely link manufacturing execution systems (MES) into this same smart factory concept. That concept for modern factories is also discussed later in the study as a part of the "factory of the future".

Meyer et al. (2009, p.1) mention enhancement of the key location factories as one of the main motivations for manufacturing development in the era of outsourcing and off-shoring. With these developed factories and cheaper and faster logistics it is possible to make much more flexible supply chains to serve the market needs, when compared to those where manufacturing is situated in the cheap labour countries of the third world. Another source of motivation for manufacturing development is the vision that manufacturing industry in general works as a solid foundation for the national economy (Johanssen et al. 2010, pp. 11-13). Therefore also the high wage countries such as Nordic countries have an underlying need to find new ways to keep at least some of the manufacturing local (ibid).

1.3. Research method and approach

The study is split into theory part and empirical part. The theory part of the study is done as a literary research, and therefore follows a conceptual, descriptive approach. The empirical research is made with action-oriented approach.

Therefore the results of the study are qualitative instead of quantitative, as there are no measurements on the effectiveness of ubiquitous technologies on manufacturing, but instead a framework of implementing ubiquitous manufacturing requirements into PDM. The last parts of the study also include some discussion on its effects on manufacturing.

1.3.1. Fitting the study into Hevner’s model

The research method used in the study is also in compliance with the model presented by Hevner et al. (2004), although a stripped, “light” version of the Hevner’s model is used. It should prove to be more suitable for a study with the depth of a master’s thesis. The framework for information system (IS) research represented by Hevner et al. (2004, pp. 78-81) focuses on the alignment of IS with organization’s business strategies & goals and also organizational infrastructure. However, as this study for prototype system does not have a clear “customer”, the strategy of a company is treated simply as “profit”, and IS strategy as “gain more information in order to update processes to make more profit”. The organizational infrastructure is simplified into generic company model that only includes the stakeholders that are in direct contact with manufacturing process of the company (such as materials purchasing). This means that the environment of the information system is not thoroughly explored and taken into consideration. This is because the ubiquitous concepts discussed in the study are mostly related to internal functionality and performance of a company. For visualization of this model, see fig. 1.1.

Due to the limitations of the study, the model and concept application part and knowledge addition part of the model are left for smaller consideration, as the focus is on developing new models. Therefore the main results of the study are these models and concepts and their minute evaluation, but the actual applicability to the environment is not tried out further than the test case. The addition to the knowledge base is also rather limited, as the study does not aim to create new knowledge, but is more focused on applying it to fulfil business needs instead.

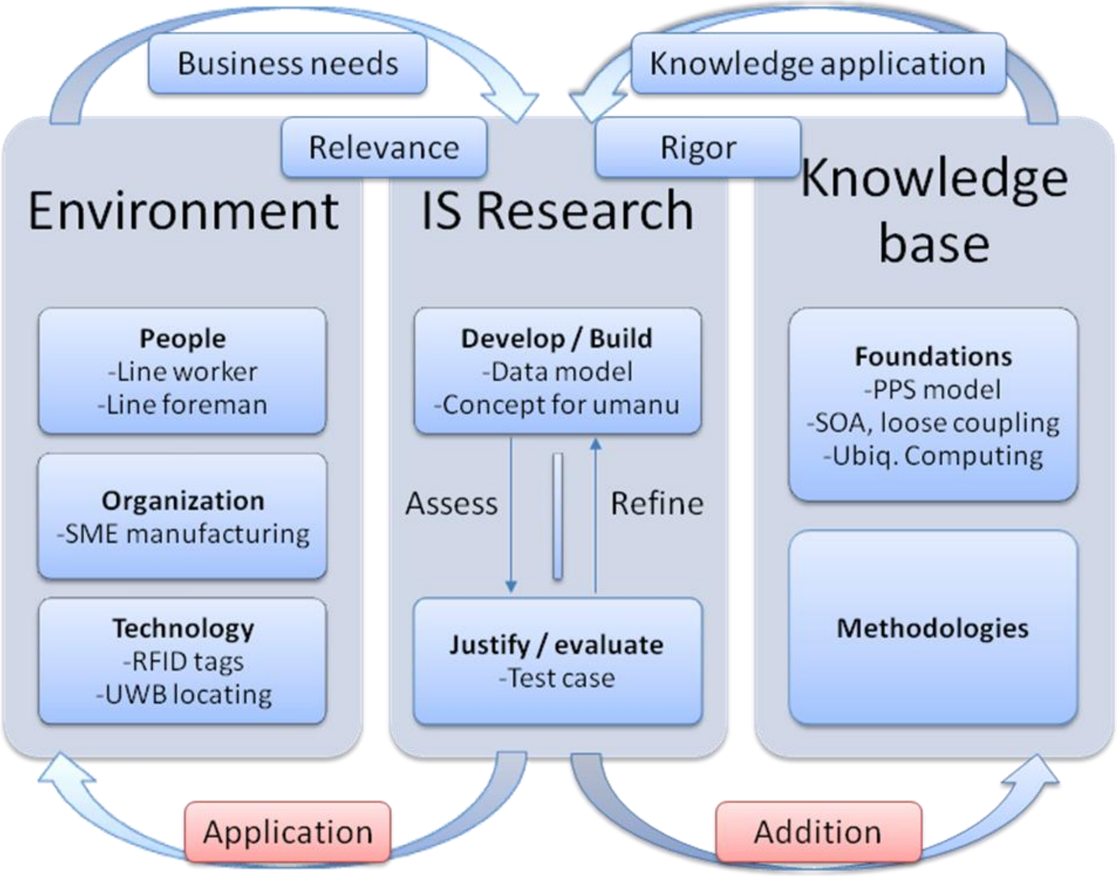


Figure 1.1: Topics of the study fit into Hevner's information system research framework. The model adapted from (Hevner et al. 2004, p. 79)

Besides from the model, the study is attempting to relate to 8 guidelines of design-science research, as presented in (Hevner et al. 2004, pp. 83). Short description of the guidelines can be found in list 1.1. This is done in sense that the guidelines presented by Hevner et al. (ibid.) are too heavy to be followed rigorously for the scope of the study, but nevertheless present a good practice and a checklist to lean on.

1. Design as an artefact – Produce a viable construct or model
2. Problem relevance – Study needs a relevant business problem to solve
3. Design evaluation – Utility of the artefact must be evaluated
4. Research contributions – Research has to able to contribute something to area
5. Research rigor – Research is done methodically and rigorously
6. Design as a search process – Artefact is functional in problem environment
7. Communication of research – Research is presentable

List 1.1: Guidelines of design-science research (adapted from Hevner et al. 2004, p. 83)

1.4. Structure of the study

The thesis consists of theoretical part of chapters 2 to 5, which aim to clarify underlying terms and to answer to research questions mainly through literature research. The basic theory and concepts of ubiquitous computing and PDM are situated in chapters 2 and 3, where chapter 4 starts to mix together the concepts and theory presented in chapters 2 and 3. Chapter 5 rounds up the theoretical part and works as a summary.

The main content of the empirical part of the study is the assembly of a product data model that supports ubiquitous manufacturing in chapter 6. This data model and some ubiquitous manufacturing concepts are also briefly validated with a pilot production line in chapter 7. The pilot production line should also help to generate new requirements and ideas regarding the topic and further gain insight into research questions. The completion and results of the case study are further detailed in chapter 8.

Chapter 9 is a conclusion for the study, wrapping up the empirical content and discussing its implications. To further illustrate the structure of the study, see fig. 1.2.

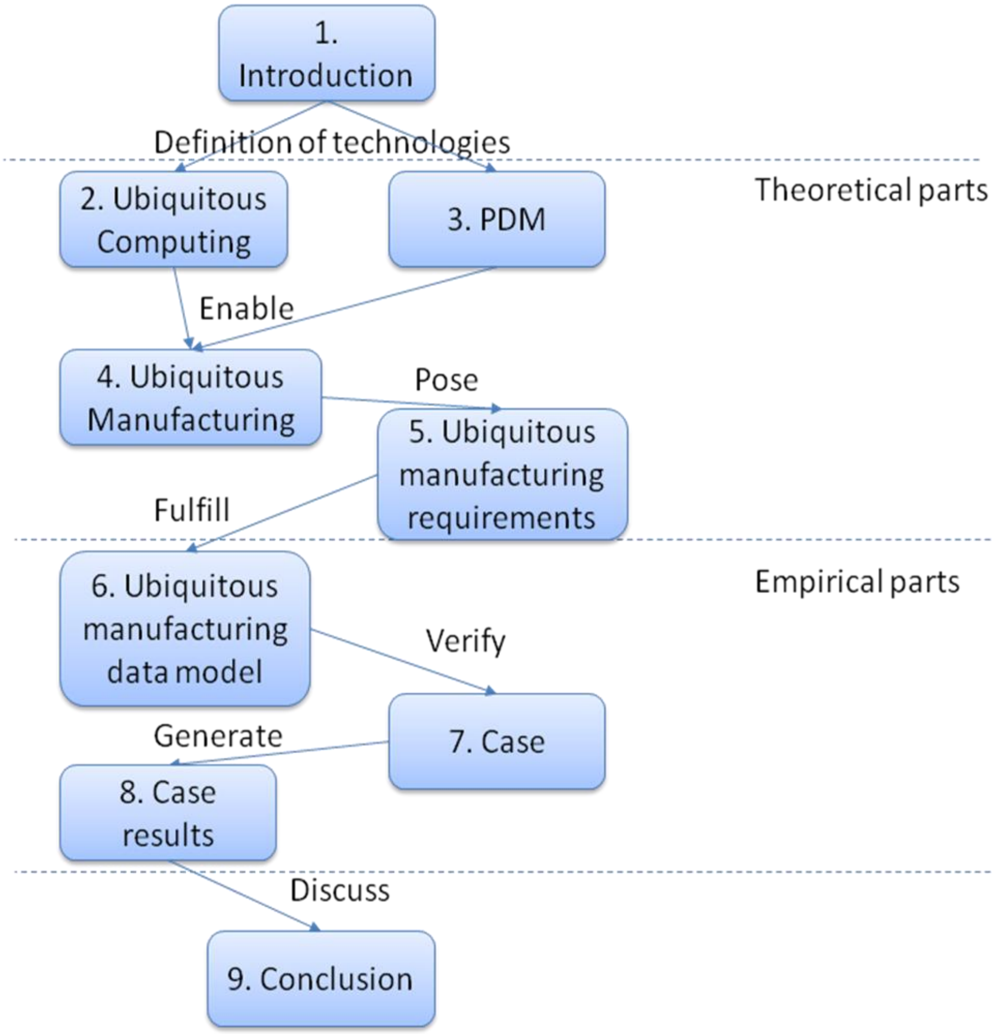


Figure 1.2: The structure of the study and the relations of the chapters

2. UBIQUITOUS COMPUTING

Ubiquitous computing is an ideology that aims into networking of everyday items in order to make lives of users easier. The term “ubiquitous computing” was originally coined by Weiser (1991), defining that the idea of ubiquitous computing in general is to produce an internet of things by having items contain information, and communicate that information to each other. This is the significant difference to the “classic” computing where the communication is mainly between the (human) user and the system. A ubiquitous computing system is not only meant to communicate and transfer data, but also to process the data more or less autonomously in order to create “intelligence” (Weiser 1991). Diegel et al. (2004, pp. 168-169) emphasizes that ubiquitous computing includes both more seamless communication/interaction of human to machine and machine to machine. Other synonymous terms commonly associated with ubiquitous computing are pervasive computing and ambient intelligence.

The ubiquitous technologies have not received too much interest in media earlier as the price of technology for producing such systems has been rather high. Now that the price and size of components has been constantly decreasing (Subirana et al. 2006, p. 13), ubiquitous computing is regarded as much more viable, although still thought as futuristic vision. Currently the most popular ubiquitous application seems to be smart home (Ma et al. 2005 pp. 56-58, Diegel et al. 2004), as it is generally used as an example when explaining the ubiquitous computing concept. This chapter aims to clarify the basic ideology and methodology of ubiquitous computing, so that it can be further implemented into manufacturing context in the following chapters. This chapter along with chapter 3 also forms the knowledge base foundation and technologies for the IS research conducted later in the study.

2.1. Ideology

The main goals of ubiquitous computing are to render computers invisible, provide more intelligent services and to lessen the burden of direct interaction from the user (Abawajy 2009, p. 71). This is done through objects and devices that offer interfaces to other devices, and communicate via them without constantly needing the attention of the human user. The intelligence also tries to adapt to human behaviour in order to be able to “guess” its users will, and work according to that.

Ma et al. (2005, pp. 53-54) defines ubiquitous computing as a bridge between virtual digital world (digital data stored in internet etc.) and the real physical world. This is then defined as “computerized physical real world” or a highly computerized “smart world” (ibid). This kind of concepts could include location bound digital content, such as digital advertisements that are shown on a PDA when near a shop or virtualized tour to a city centre, such as Google Street view.

Apart from the objects being intelligent and communicating with each other, another big tendency in the discussion about ubiquitous computing has been the human interface of ubiquitous systems. This interface is meant to be very human centric, so that information is presented to people in convenient way without distracting or being intrusive, and without needing the constant attention of the user (Abawajy 2009, pp. 61-62, Gill & Cormican 2006 p. 3). Another term about ubiquitous computing that emphasizes this human – machine interaction is “calm technologies” (Weiser & Brown in Abawajy 2009, p. 67).

On the whole ubiquitous computing wraps up few modern concepts in order to model a futuristic vision of computing. Main difference between authors in the field of ubiquitous computing seems to be the focus. Some emphasize human - computer interaction, whereas others focus on interlinked smart objects.

In this study the main focus is on the communication and intelligence of the objects, and how these intelligent objects are modelled. This means that the human - computer interaction is left to a smaller part, as that area is more from the field of human-centred technologies and usability, than of knowledge management.

2.1.1. Internet and the future of computers as a foundation

The first requirement set by ubiquitous computing is to have the connecting medium for the objects to exchange information. The internet has created a standard and a foundation for ubiquitous systems to exist (Jianhua et al. 2005, p. 53, Zuehlke 2010, p. 130). Weiser’s (1991) original idea of the ubiquitous system being accessed through any terminal (PC, handheld / PDA or even a “scrap computer”) diminishes the idea of the current modern personal computer. The internet has brought computing one step closer to this, as one can access the same web space through any terminal, whether it was handheld, full sized or someone else’s PC. Because of this, it can be argued whether the term PC is getting more and more outdated, as PCs as tabletop devices are not that “personal” anymore (Garfinkel 2010, p. 86). This is caused by more and more applications and services that are moving over to the omnipresent internet, where they are available for use through any computer, not just from the personal one (ibid.).

In ubiquitous computing sense the personal functionality can be seen to have moved to smaller mobile devices, whereas classic tabletop computers (PCs) have grown more non-personal. This has happened through the mobile devices starting to have capability for classic PC functionality, while the most personifying aspect is that in the current world mobile devices are becoming more and more inseparable from their owners in terms of location and availability.

The same approach that is present in the "depersonalization of PCs" can also be somewhat seen in recent trend of cloud computing, where computational resources are dynamically allocated on need basis instead of permanent solutions. So that there is no need to only rely on having your applications and services dwelling in internet, but also the hardware is made accessible and ever present. (Youseff et al. 2008, pp. 1-2)

Cloud or internet based system architecture establishes requirements for data availability and accessibility, as the usage of a system is distributed by default. The data has to be accessible from multiple computers and it may be accessed and modified by multiple users simultaneously. This adds the need for document and version management in the any system, which is described in more detail in the PDM environment at chapter 3.1.

2.1.2. The Internet of things

The internet of things fulfils the part of the ubiquitous computing ideology that aims to interlinking and connecting all devices to each other, via internet. This relates to the notion that the internet is not only a medium for people to communicate and do things, but one that automatic devices can use as well.

Kranz et al. (2010, p. 46) defines the internet of things as the result of embedded readiness for communication, interaction and processing between the objects. These devices or objects do not necessarily have to be computers in classic sense, but can rather be any kind of objects, such as tools, clothes or furniture (Kranz et al. 2010, pp. 47-48). Of course these objects have to be further "computerized" by adding the required technologies that allow interaction. These technologies are further described in chapter 2.2.

Classically the internet has consisted of PC clients that are attached to servers and each other via hardware infrastructure (routers etc.). This would be example of a network designated for communication and usage of people. The internet of things expands the concept of linking people to information and each other to interconnecting devices cooperating for wider variety of functionality. The same paradigm shift can be currently seen in the rise of "web 2.0", which connects users directly to each other in terms of content creation and interaction, instead of being web page centric like the internet of the 90's.

Kranz et al. (2010, pp. 50-51) also make an important notion about internet of things (and ubiquitous computing), that the key is embedding more intelligence into already existing objects, instead of creating new intelligent objects. Therefore the fundamental role of the ubiquitous computing is to enhance current functionality of objects.

2.2. Technologies & techniques

Most of the ubiquitous computing concepts are based on presence, locating or ambient awareness of surrounding objects and systems. This is awareness is usually generated through location or proximity sensing, which makes locating a critical factor in forming ubiquitous system (Westkämper & Jendoubi 2003, p. 3, Diegel et al. 2004, p. 169). Communication between objects is usually done through wireless technologies, such as WLAN, Bluetooth, RFID or radio signals (ibid.). Some commonly used location and positioning technologies are GPS, UWB and ZigBee. Other used technologies include machine vision applications and the usage of wireless communication technologies, where their limited range can be used to pinpoint locations of objects.

The main differences between these technologies are their effective radius, topology and power consumption. WLAN technologies require an infrastructure of wireless network (usually generated by a router), where they can communicate with other clients. Also if a WLAN router is connected to a larger network, the clients can seamlessly contact other objects and computers in the whole network as well. RFID clients are just tags that contain information that can be read or written wirelessly by a reader (Westkämper & Jendoubi 2010, pp. 3-4). Therefore the RIFD clients, tags, cannot contact each other, but instead a RFID reader can relay the information contained in the client forward. The main advantages of RFID technologies relate to them being cheap to manufacture, and the possibility to have tags / clients that do not require power (passive tags) (ibid.). Bluetooth is another commonly used technology, where the clients communicate with each other through radio waves. The difference to WLAN is that Bluetooth does not need an established network or a router, so the Bluetooth devices can communicate with each other anywhere within the range of each other.

For enabling ambient awareness, locating, identification and communication are usually enough (Haller et al. 2008, p. 3). The objects have to know their location either absolutely or relatively to other objects, and they must be able to react according their location and state, and when needed communicate their plans or requests to other objects in the system.

Diegel et al. (2004, p. 169) discuss that the “intelligence” produced by a network of objects can be formed by the sole functionality of every single object and their transactions, or contained in a central computer that controls all the devices, and their interaction. The main difference in these two approaches is the autonomous functionality of the first, where no central computer is needed to govern the action. Then again it can be argued that the central computer more easily enables controlling the “big picture” of the whole system. This architectural issue is further discussed in chapter 4.2.

Embedding the computers and logic into devices has become available to wide use through the constant miniaturization of microchips and the declining prices (Ma et al. 2005, p. 53). Putnik et al. (2007, p. 3203) also points out the affordability of high performance and low power electronic components that can be wirelessly connected, as drivers for new computing concepts. Some components can even harvest the energy from their surrounding environment (ibid.). This combined with components that even do not need power, such as passive RFID tags has made ubiquitous computing more and more realistic vision of the future.

Also the widespread accessibility of internet makes it possible for the objects to communicate with each other in real time. Another big enabling technology for forming an internet of things is the upcoming IPv6 standard. IPv6 multiplies the address space so that virtually all devices connected to internet can have their unique address, and be much easier to communicate to (Haller et al. 2008, pp. 9-10). Current IPv4 cannot support the sheer amount of addresses that are required by the devices currently (ibid.). The problem of running out of addresses is currently solved by “hiding” multiple devices into private networks under single IP address with NAT (Network Address Translation). All the objects in the network need unique identifier and an address, so that they can be reached when needed (ibid.).

Haller et al. (2008, pp. 2) also remind that the technologies to manufacture such ubiquitous systems will surely be subject to change in the coming years, but the concepts, goals and ideology of ubiquitous computing should be more constant.

2.3. Summary

The key meaning of ubiquitous computing for ubiquitous manufacturing is to provide the ideology and means for interconnected, smart objects. These smart objects can make lives of users easier through automating actions and allowing new kinds of interaction through events and real world actions, such as for example entering a certain area. The smart objects themselves do not need to be brand new devices or objects, but old objects that are enhanced with tags or micro-sized computers for ubiquitous functionality.

The same ideology is ported into manufacturing environment later in the study, as a part of the ubiquitous manufacturing concept. Clearly technologies for enabling autonomous connected objects already do exist. The standardization seems to be missing though, as there currently are very few consumer level applications available.

3. PRODUCT DATA MANAGEMENT, PDM

This chapter aims to clarify what is product data, and what are the underlying trends and issues in product data management. Product data management is mainly inspected from manufacturing point of view, as that is the scope of the study.

The chapter also gives the reader the idea of current PDM systems and their functionality, which works as a basis for ubiquitous manufacturing data management discussed in following chapters. Also the reasons why PDM systems are taken as a foundation for ubiquitous manufacturing are further detailed in chapter 5.

3.1. Basic concepts of product data

Product data means all the data and information related to the products and the offerings of a company. A product itself can contain a physical, tangible article, a service and information (Peltonen et al. 2002, p. 12). Usually the products that are sold form a mixture of these three components. For example carton of milk would be mostly just the physical article, whereas a book would be bought mostly (if not only) for the information. A haircut is a simple example of a product that is entirely a service. A mobile phone as a product would also include lots of information in addition to physical article, but there could also be services embedded to it too, like a long warranty or free software updates. But even if the products are simple, there still might be lots of information regarding the process of manufacturing and delivering the product. As the context of this thesis is manufacturing, we’re mostly focusing on the physical articles as products, and the information contained in them and the processes to create them. The service aspect of the product is therefore left to a more minor role.

The need for PDM systems to manage the product related information rises from the ever increasing complexity of products, where a single person can no longer handle all data needed and present in a single product (Kropsu-Vehkaperä et al. 2009, p. 759). An example of this would be a mobile phone, where lots of various technologies are present, regarding the physical construction, software and communication protocols are present. But even simpler products such as groceries may contain increasing amounts of regulations and intricate processes for manufacturing.

The scope of PDM systems can vary very much from system to system, and is usually heavily linked on corporate ERP system (Peltonen et al. 2002, pp. 10-11) and to the complexity of the products. In optimal scenario the PDM system should nevertheless be able to hold all information regarding the product portfolio. This information includes all information needed throughout the lifecycle of a product, ranging from designing to recycling and disassembly.

Peltonen et al. (2002, pp. 10-11) point out document management, product structure management, change and version management and title/item management as the main elements of PDM. These elements are further detailed in chapters 3.1.1 through 3.1.4. Kropsu-Vehkaperä et al. (2009, p. 760) also adds configuration management, information warehouse, workflow & process management and system administration management as the essential, basic modules of a PDM system. The essential PDM modules suggested by Kropsu-Vehkaperä et al. (ibid.) are more functionally oriented and / or support functions, whereas PDM elements by Peltonen et al. (2002, pp. 10-11) describe the database and general requirements for PDM.

3.1.1. Title / Item management

Title management is the fundamental part of PDM, as it governs the naming conventions, terminology and product lines. The title management essentially answers the questions about what are the manageable objects, and how they relate to each other and the whole manufacturing process, what are the categories of titles and how the title hierarchy is formed (Peltonen et al. 2002, p. 45). It is also important to decide what information is stored about all the titles, and how the attributes and the data are formed. Peltonen et al. (2002, pp. 15-16) define that successful title management is required before even starting to plan other functions of PDM, which makes it a foundation for any PDM system.

3.1.2. Product structure management

The product structure straightforwardly means the hierarchical structure of the components of the product. A simple product structure can be visualized as a tree shaped data structure, where nodes of the tree are components (subassemblies), leaves are subcomponents and the root is the finished product. See fig. 3.1 for an example of this. In the structure only the leaves (items furthest from the root, final product) are atomic components, where all the nodes that connect to multiple leaves are subassemblies. In the ballpoint pen example outer casing is a subassembly that is formed by a plastic tube and metal tip frame, which are atomic components (parts). A bill of materials (BOM) in this case would only contain the atomic components.

To further help illustrating ballpoint pen examples used through the study, see detailed structure of an example pen in appendix A.

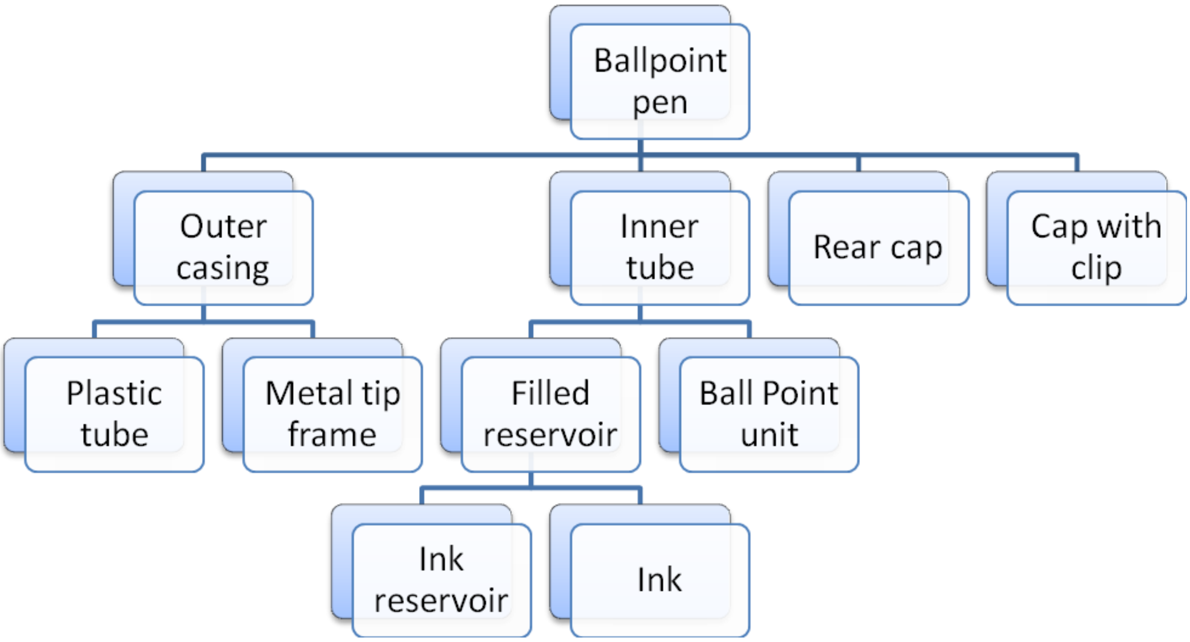


Figure 3.1: A simplified product structure tree

A simple product structure, such as one shown in fig. 3.1, does not take into account how the assembly is done, as it only states the components of the assembly, and partially the order of assembly (as it is not possible to hop over nodes when moving from leaves towards the root). Product structure also does not usually contain any version management, and modelling configurable and customizable objects can turn out to be cumbersome as well.

A single product can also have multiple different structures, such as component structure, area structure, functional structure or electrics and cabling structure for an example. In cases like this it is important that these structures can also be cross referenced where applicable. (Peltonen et al. 2002, pp. 64-66)

3.1.3. Document management

Besides the physical structure of the product, product data contains documentation, ranging from development decisions and their justifications to manufacturing instructions, manuals and marketing materials. Almost all the content on a PDM system can be seen as documents, so naturally document management is needed to make them accessible, easy to use and conflict-free.

Document management also includes the states of product data, as some documents might be waiting for approval, and some might be locked or frozen so that nobody else can commit changes to them simultaneously. All updates into products and documents have to range through the whole system, so that no data is left non-updated. Saaksvuori & Immonen (2008, p. 8) also remind about the importance of metadata in a PDM system. This includes the description of the product data including what kind of information it is, where is it stored, who has recorded it and how can it be accessed (ibid.).

3.1.4. Change and version management

Many products still undergo development even after they have been put to manufacturing, which adds to the need to make changes in product related data. The product data also needs to be available for each version for possible maintenance and backtracking cases (Peltonen et al. 2002, p. 77). Also the whole development process needs a platform as well, as a fundamental part of the product data and documentation is generated during development. It is also worth notice, that during development the product data is subject to change a lot, so the change and version management play a crucial role there. Of course all the development data cannot be mixed with the actual production, but having the same system could make the data from development to production transition easier, while harmonizing the data.

3.2. PDM systems and standards

Saaksvuori & Immonen (2008, p. 10) remind that product data management is mainly carried out by various information processing systems, as only simple product data management is possible without IT nowadays. This chapter discusses these systems and their integration to other areas of the enterprise IT infrastructure.

Peltonen et al. (2002, pp. 105-109) points out manufacturing ERP and design CAD systems as the most important points of integration where PDM system is connected to other enterprise functions. There does not seem to be any dominant PDM systems, but many authors (Lanz 2010, Peltonen et al. 2002, Sackett & Bryan 1998 etc.) name STEP as an example when giving examples and talking about PDM standards. This easily leads to believe that companies either do not have dedicated PDM systems, or that they have various non-unified tailored systems. Peltonen et al. (2002, p. 93) also reminds that even STEP system is not used too widely, but only some parts of the STEP standard are used in PDM systems. More about STEP standard can be found in the following chapter 3.2.1. Lanz (2010, p. 9) also criticizes that the impressive number of models and their extensions cripples the idea of one model emerging as a dominant standard.

Ou-Jang & Chang (2006, p. 369) also argue that the relationship between ERP and PDM systems can be problematic, as both can contain some overlapping functionality.

The main questions in system integration are which of the systems works as a master, having the responsibility on data and if the integration is only one way or two way (bi directional) (ibid., Peltonen et al. 2002, pp. 107-109). In some cases with older ERP systems the PDM integration can be very difficult if not impossible (Peltonen et al. 2002, pp. 107-109).

3.2.1. STEP and EXPRESS

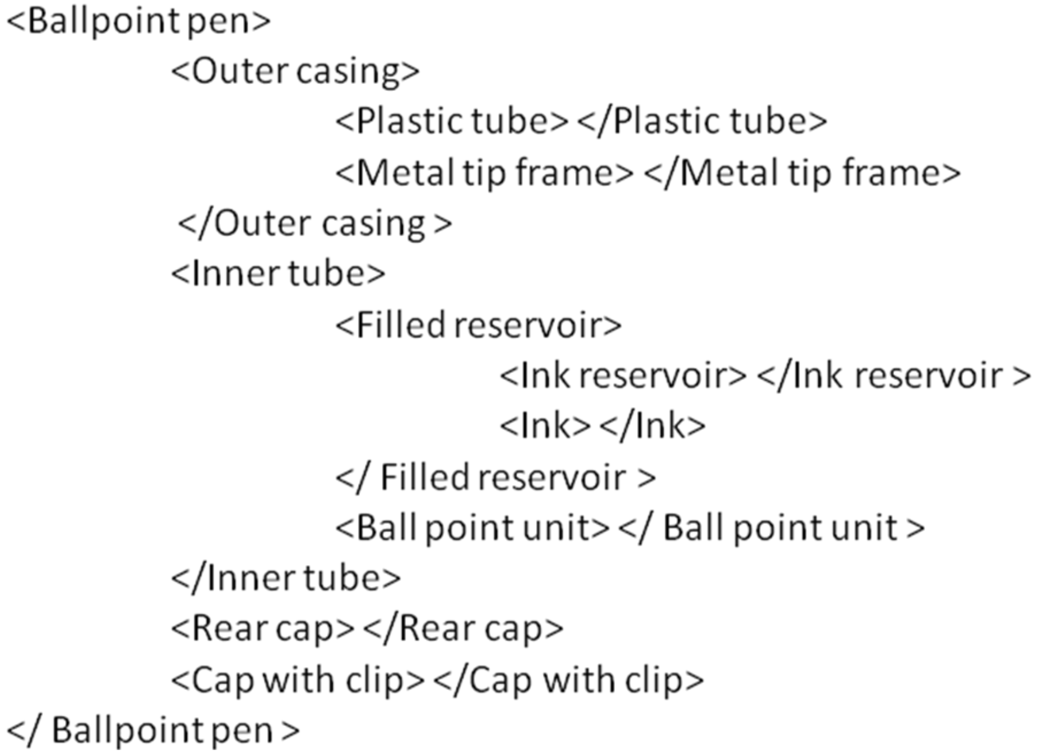
STEP (Standard for the Exchange of Product Data) is a collection of various ISO standards for forming product data in certain industrial sectors (Peltonen et al. 2002, p. 93). STEP is also known as ISO 10303, which is the official name for the standard. All STEP models are formed with EXPRESS language, which is an object based modelling language. According to Peltonen et al (2002, p. 93) STEP was originally planned for transferring product data from one PDM system to another, but currently it is mostly used as data model in PDM systems.

Peltonen et al (2002, pp. 93-95) criticizes that systems are not really compatible with each other by just being STEP compliant, but they have to have the very same application protocol as well. STEP protocol consists of hundreds of application protocols for various different lines of manufacturing, and single protocols can be very heavy to use (manuals of hundreds of pages). The most used of STEP standards are geometric data formats use for transferring data between CAD systems. (ibid.). Lanz (2010, p. 24) also denotes that core STEP format has not fit into companies too well, so that new expansions and add-ons were and are constantly built, further moving away from standardization.

3.2.2. XML

The extensive mark-up language (XML) standard is also used in PDM, but it is actually just a scripting language for encoding documents in machine-readable form. This means that no matter the PDM system, the documents can be formed as XML (Peltonen et al. 2002, p. 102). XML can therefore also be used for text form messaging between integrated systems, such as ERP and PDM (Peltonen et al. 2002, p. 109).

XML documents consist of elements that can be also situated within other elements to form hierarchies. XML works specifically well with product structures, as XML documents can also contain the same conceptual representation of a tree data structure used in simple product structures. See the example of the product structure shown in chapter 3.1.2 as simple XML document in fig. 3.2.



```
<Ballpoint pen>
        <Outer casing>
                <Plastic tube> </Plastic tube>
                <Metal tip frame> </Metal tip frame>
        </Outer casing >
        <Inner tube>
                <Filled reservoir>
                        <Ink reservoir> </Ink reservoir >
                        <Ink> </Ink>
                </ Filled reservoir >
                <Ball point unit> </ Ball point unit >
        </Inner tube>
        <Rear cap> </Rear cap>
        <Cap with clip> </Cap with clip>
</ Ballpoint pen >
```

Figure 3.2: A simplified product structure in XML

3.3. Product data for manufacturing

Product data has usually numerous different views depending upon the part of organization that is inspecting it (Kropsu-Vehkaperä et al. 2009, p. 765). Designing, manufacturing and sales they all might have varying idea of the product, and all of them require different information about the product. Product data for manufacturing naturally emphasizes the data needed for manufacturing the product, such as bill of materials (BOM), manufacturing bill of materials (MBOM), and instructions needed for assembly (such as documents, machine and process instructions or CAD diagrams). Saaksvuori & Immonen (2008, p. 8) also remind that these BOMs and MBOMs are usually just flat lists of components, and does not include product structure, assembly or component hierarchy. Apart from BOMs these other instructions for manufacturing might include for example processing times, treatment rules, raw material quantities and qualities and processing directions and rules. Also some metadata of the components is usually needed, such as ordering and billing information of bought components, and manufacturing phases and routing of self made components.

Kropsu-Vehkaperä et al. (2009, p. 770) also remind that the importance appropriate product data and PDM is even higher when manufacturing complex products, where lots of various information has to be available in orderly fashion.

The data needed for assembly and manufacturing varies from process to process (Kropsu-Vehkaperä et al. 2009, p. 765), as the processes might not need to know the data of other processes. This means that usually only a small amount of the vast product data is needed by a single user, such as operator of a manufacturing process. Naturally some product data such as user and marketing manuals, some of development & testing data and alike are not likely to be needed in manufacturing phase.

Besides product related data, the manufacturing also needs information about the resources of the manufacturing plant, such as their availability and allocation, and the timetables for outstanding orders. This is another point of integration between PDM and operative ERP or management executive systems (MES). Kovacz et al. (1998, p. 285) criticizes that PDM systems are seldom used beyond design process, and manufacturing phase has its own work flow management systems. Nevertheless the integration between PDM and work flow management is agreed to be beneficial in many cases (ibid., Kropsu-Vehkaperä et al. 2009, p. 759).

3.4. Product lifecycle management, PLM

Another closely related modern trend of PDM is to manage the whole lifecycle of information about a product from its design to manufacturing and all the way through usage to discarding and recycling of the product (Kropsu-Vehkaperä et al. 2009, pp. 760, 763). The term product lifecycle management can be a bit fallacious, as the idea of PLM is not to manage the lifecycle of a product, but to manage the whole product and all its related information through its whole lifecycle. Therefore it can be expressed that PLM is a concept enhances the concept of PDM to cover the whole lifecycle of a product, instead of focusing just to manufacturing. Ultimately the both terms are rather ambiguous, and eventually mean more or less the same thing. This is the case also in this thesis, as the PDM system conceptualized here is not strictly limited into any certain phase of product life, even though the focus is on beginning of life (BOL).

The part of PLM that relates heavily to manufacturing is the seamless lifecycle management up to until the product is ready. Saaksvuori & Immonen (2008, p. 2) defines PLM as a holistic business concept to manage a product and its lifecycle including items, documents, BOMs, analysis reports test specifications, environmental component information, quality standards, engineering requirements, change orders, manufacturing procedures, product performance information and so forth. Generally there seems to be agreement that PLM is more of a strategic approach, than just another IT system (Kropsu-Vehkaperä et al. 2009, p. 760, Sudarsan et al. 2005, p. 1399). Another way to draw a line between PLM and PDM is that PDM covers “static” product data, such as specifications, instructions and BOMs. On the other hand PLM is used to cover more dynamic product data, which occurs during distribution, usage and end-of-life of the product. (Kropsu-Vehkaperä et al. 2009, p. 760)

Saaksvuori & Immonen (2008, pp. 2-3) also state that although the core functionality of a PLM system is the creation, preservation and storage of product related information, a modern PLM system also includes functional capacities such as a workflow and program management and project control features for product management operations.

PDM (and PLM) can also be used for any other product related data collection as well. Luh et al. (2010, pp. 223-224) point out that PDM is a viable place to collect data about the environmental impact of the products related to materials used in them. This way it is possible to clearly point out the environmental effects of single components, and their cumulative effect on a single product.

3.5. Summary

The field of product data management is currently rather fragmented, in sense of multiple various systems, but also in sense of system responsibilities, as different PDM systems can have very different functionality. Nevertheless the main priorities of current PDM systems seem to be to help product development and supply manufacturing with the information how to make a product. These functionalities are facilitated by title, product structure, document and change management offered by PDM.

PDM system also facilitates lots of data required by other systems of the company, and therefore can be seen as an integral component, with high priority on availability and integration capability. Another apparent need for PDM also seems to be the possibility to integrate product data usage during the birth of life (BOL) of the product, especially in design and manufacturing. More discussion about this gap present in product data usage to come in following chapter 5.1. Recent interest in PLM systems could also indicate that possibly a shift towards more dynamic and functional PDM is bubbling under, instead of just the static view to product data.

4. UBIQUITOUS MANUFACTURING

This chapter discusses about ubiquitous manufacturing (later referred as UM), which straightforwardly means the usage of ubiquitous computing concepts in manufacturing environment. The concepts discussed here provide requirements for the product data management model to be developed in the following chapters.

Apart from facilitating ubiquitous computing concepts in manufacturing, another integral idea of the study is to explore how the PDM functionality could be enhanced in manufacturing through more flexible data management. This is done especially in sense of the PDM needs to integrate product data usage during design and manufacturing.

4.1. Definitions

The term “ubiquitous manufacturing” is not yet very well established, and because of this numerous authors have their own definition and concepts for the term. In this chapter some of these definitions are presented and discussed.

Lucke & Constantinescu (2010, p. 1) place the main focus of ubiquitous manufacturing on context aware smart factories that assists people and machines in execution of their tasks. Some ubiquitous manufacturing concepts focus on collecting information through the entire lifecycle of a product and using it for design and manufacturing (Suh et al. 2008, p. 542). Another futuristic vision for ubiquitous manufacturing is enabling automated unmanned factories that can automatically assemble and configure orders according to customer’s wishes, while giving real time information about progress of the order to the customer (Diegel et al. 2004, pp. 175-176).

Although varying a bit, most of the concepts regarding ubiquitous manufacturing seem to focus on data collection and automation through ubiquitous technologies, which should give a rough definition for the concept. Putnik (2010, pp. 1-2) discusses about whether ubiquitous manufacturing means the ubiquitous availability of the manufactured goods, or the schema of using ubiquitous computing –like manufacturing processes. In this thesis, the idea of ubiquitous availability of goods is bypassed, as the focus is on enhancing manufacturing, not on availability and logistics.

4.1.1. Ubiquitous computing in manufacturing context

Ma et al. (2005, 53-54) expressed their vision of ubiquitous computing enveloping the whole world with smart interacting objects, but the manufacturing concept does not really require such global standardization. For manufacturing to use ubiquitous functionality, linking a single factory or a part of the supply chain can suffice. This also makes manufacturing very viable ground to use ubiquitous technologies, as factories and processes are rather standard processes where the interfaces are possible to digitalize. The spatially closed nature of manufacturing plants also makes it easier to implement spatially aware ubiquitous intelligence in to the site.

The main vision of this study is forming the internet of things, as defined in chapter 2, in an enterprise context where the production elements are smart objects that communicate to each other. The communication of these autonomous objects allows them to become active participants in business processes, and bridge the gap between physical world and world's representation in information systems (Haller et al. 2008, p. 2). This approach to ubiquitous manufacturing does not limit the applicability of the ubiquitous functionality in a production plant, but can be seen as a foundation for many of the concepts described later in this chapter.

4.2. Data architecture

The ubiquitous concept adds some new elements into data that is used for manufacturing. These elements are discussed in this chapter, as well as the requirements for the architecture to smoothly handle the new data.

4.2.1. New data needs caused by individuality

In order to bind the ubiquitous interactivity into objects and processes, the individual information has to be generated first, and then embedded or linked into these objects and processes. Lucke et al. (2008, p. 116) describe object identification, positioning and status information as the three most important factors in their smart factory concept. All of these are instanced data, which is specific to each individual object in the system. This means that the location, identification, context, state and interface of all the key processes or objects have to be stored and available when referenced with the ID of the object.

Naturally if all the objects in a factory are regarded as individuals in digital context, the amount of data compared to current systems multiplies. Of course not all the identical product related data has to be stored about every single object, but only the instance related data. Instance related data could consist for example from unique ID, state, location, state and possible custom orders and flags. Including all the object or product related data into every single object, such as detailed specifications of identical objects, leads to duplication and distribution of data, which are generally unwanted properties from a database system.

4.2.2. Functional architecture: central computer or autonomous objects

One of the main choices that have to be made when designing a ubiquitous system is the architecture, or network topology, of the system. The choice is whether to make objects interact only with each other, only with a central computer or something between these two extremes (Satyanarayanan 2001, pp. 6-7). In networking server-client architecture this has generally been the discussion of thick client (client does most of the work) or thin client (server does most of the work). In the ubiquitous sense the autonomous objects that interact directly with each other can be seen as thick clients and server-oriented system without direct interaction between objects as a system with thin clients (ibid.).

Also in ubiquitous sense there is a need for central computing to get a high level picture of the systems actions. This means that the central computer can work just for logging and monitoring. On the other hand objects can be very lightweight too, so that all their instanced information is stored in the database, and they only contain RFID tag or alike for the identification (thin client). In this case when a lightweight object enters a process, process checks the tag and searches the database for the object, and processes it according to flags and orders in its data found from database.

From manufacturing point of view it seems to make more sense to have central computer instead of fully autonomous objects that interact with each other. This is mainly because of the stabile environment of a manufacturing plant, and the fact that central computer allows easier access into all data regarding the objects in the shop floor. With central computer the lightweight objects do not need to do processing capabilities and can be formed with low power or even powerless solutions, such as passive RFID tags. Central computer architecture with light objects works also in harmony with SOA ideology, where the central computer acts as registry pointing objects to each other. More about SOA is in the following chapter 4.2.3.

With central computer it is also easier to reduce the amount of complexity (and costs) of implementing ubiquitous technology into objects, as they do not really have to communicate between each other, but only with a central computer that works as a middleman. This means that when an object moves in the shop floor through various stages where it has to communicate with other objects and processes, it does not need to know the addresses or the exact interfaces of others, but only the central computer. Also the single interface architecture makes it easier to manage objects in order for future modifications and updates. Then again some researches (Lucke et al. 2008, p. 115, Westkämper & Jendoubi 2003, p. 1) clearly prefer to have the structure decentralized in order to be robust, as for an example such system will not need to be brought down by the downtime of the mainframe.

The system knowing the state & location of its components is more “traditional” approach to ubiquitous computing, as the system can act depending on location of the objects without other direct interaction. However the knowledge contained by objects, and objects interacting with each other can further increase the possibilities of the system automation, where objects can interact and communicate truly autonomously. Yet this approach is still more complex and without doubt much more expensive than thin clients approach.

4.2.3. Service Oriented Architecture, SOA

The Service Oriented Architecture by itself is not a concept of ubiquitous computing, but instead ubiquitous computing and especially manufacturing concepts can possibly benefit from the usage of SOA elements in forming of a ubiquitous computing system. The basic functionality of SOA is that systems or objects offer their functionality as services, whereas the parts of the system that need the functionality request these services from the providers (McGovern 2006, p. xxii). The registry works as a “phonebook” that collects the provided services, and where the clients requesting services go to find the correct service (ibid.).

This means that the interconnected objects present their service interface to each other, can function together accordingly. See a simple diagram of SOA architecture in fig. 3.1.

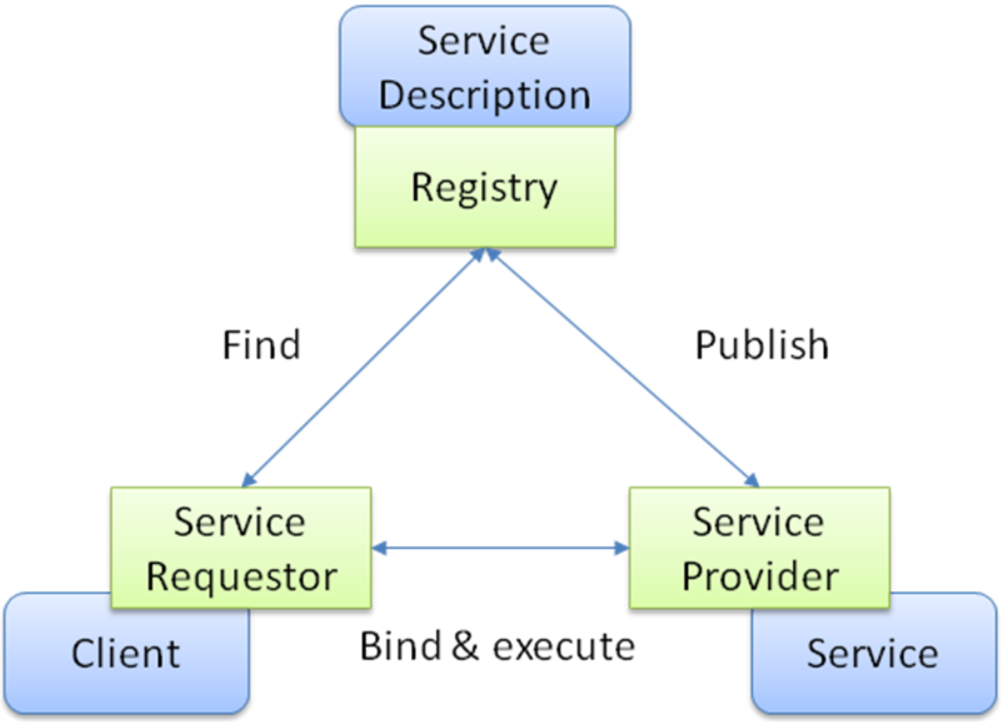


Figure 4.1: SOA architecture model, adapted from McGovern et al. (2006, p. xxii)

In ubiquitous computing sense SOA could be manifested so that interconnected objects offer their services to registry, which is then used by other objects to find the service providers. This would mean that for example the manufacturing plant processes offer their processing as services, which are then consumed by objects needing processing. McGovern et al. (2006, p. 5) also remind that SOA can work as an architecture for collecting and wrapping the interfaces of other (legacy) systems and therefore is quite applicable to various existing systems.

Zuehlke (2010, p. 135) also points out that SOA paradigm enables decentralized structures and lowers hierarchical structures. In ubiquitous manufacturing concept this could be manifested as autonomous, self-organizing objects and processes.

4.2.4. Ideology of loose coupling

The term “loose coupling” in computer science means that the objects (elements, modules or classes) do not necessarily know each other absolutely, but they only use a “hazy” interface of each other that encapsulates the functionality of the other elements (Haller et al. 2008, p. 12). With such architecture it is possible to easily add and replace components without breaking the functionality of the whole system. Also the interfaces do not have to be explicitly defined for every single object, but to use wider standards for similar objects.

The problem here yet again is the lack of standardization for a widely accepted standard, but the ideology of loose coupling is most likely a good idea to keep in mind when designing a new ubiquitous system. Loose coupling itself can be used to aid with standardization, as the existing non-standard systems can be wrapped into interfaces that can be then loosely coupled.

4.3. Real-time location systems and real data

Real-time location systems (RTLS) are special locating systems that are designed to reveal the location of the tracked objects continuously in real time. This real-time information can be used so that the system is constantly aware of the location of its objects, but also in order to make the object is aware of its own location. The locating systems can be used so that the components or processes can “advertise” themselves, their location and services they can offer.

When components know their exact location and their desired goal, it is possible for them to navigate themselves without conventional proximity sensors or machine vision in order to avoid crashes. Kaiser et al. (2003, p. 3) also mentions that these conventional sensing systems are presently poor on rapidly and accurately detecting variable obstacles in free environment with lots of variables. Ubiquitous system with RTLS can partially skip this problem, if it is possible to track and be aware of most of or all the objects (and obstacles) in desired areas. Then it is possible to navigate just by using model of the area and location & orientation of the object.

The precision of RTLS systems might set some restrictions, as it is not always possible to get the location accurately, but some processes might require very accurate precision. An example of this kind of process could be feeding metal plates to a drill that makes holes into specified spots of the plate. Naturally if the RTLS system cannot guarantee enough precision, it is possible to use a hybrid system where crude level navigation is done with RTLS, and the precision locating is done with proximity sensors or machine vision.

Where locating might have some problems with the accuracy, it is easy to do crude level navigation to help a human navigate. For example when trying to find certain components from a warehouse, the computer could mark the spots on map and show route to them akin to a car GPS navigator. The positioning systems and location information itself can also be highly advantageous in “space intensive” phases of production, such as storage and warehousing.

Location systems that work in real-time also require real-time data management throughout the system (Lucke et al. 2008, p. 116). With poor real-time data management the advantages brought by real-time systems could be consumed by excessive response times or slow updating.

Apart from locating the objects, the state of the objects was one of the critical factors discussed earlier (ibid.). If some processes change the state of certain objects, then they should add the changes also to database where the instanced data of the objects is stored, or into RFID tag that is attached to the object. This way it is possible to keep track of the progress of the products through state changes, and avoid misplacing the objects if their state is forgotten (Lucke & Constantinescu 2010, p. 3). Naturally if the aim is to digitalize all the information regarding the product, then its state should be taken into consideration as well, as it can be used as a parameter for further processes. The states of objects are also key factors in plant activity planning and monitoring, where the system needs make queries such as are the resources available or reserved.

With different locating systems, the purpose of the system should be aligned with the functionality. A trivial example of this is that there is no need to have locating service for objects that do not move, such as heavy machinery. But further refined, the objects that only move on rigid product line will not be likely to need a positioning service for the whole plant, such as UWB tags. For product line a more convenient positioning could be RFID with multiple low power readers that can only read tags from objects that are at very close range. This low power RFID reader network could then only note at which station a single product is at current time, or where was the product last seen. The more dynamic locating for larger areas would be then more suitable for objects that actually have the possibility to wander around the larger areas, such as tools, kitting wagons and people.

4.4. Ubiquitous manufacturing concepts and benefits

The key benefit of the ubiquitous technology here is the digitalization and availability of more product data than before. The increased knowledge of the status of objects in the manufacturing plant naturally opens up possibilities in process automation and better visibility through transparency (Lucke & Constantinescu 2010, p. 3). Moreover the transparency enables lots of possibilities for recording data from the processes for analysis, in order to find the bottlenecks and other critical points in the processes (ibid.).

One of the first things the transparency of shop floor enables is real time information about the phase of the production and estimates of throughput time. This naturally reflects as better tracking of orders and more timely deliveries.

Another concept present in some studies (Suh et al. 2008, pp. 547-548, Putnik 2007, p. 1) regarding ubiquitous manufacturing is that singular products support well mass customization through interchangeable parts and configurations, if the processes can support them. This is enabled by having the information of customization travel with the product in the manufacturing phase, where the customized orders can be taken when the product is at the customization point. When information about the customization is carried in, or pointed to from the customizable product, it is also easy to track the custom orders.

Another concept Suh et al. (2008, p. 546) mention their article is manufacturing routing and error recovery in case of faults in manufacturing line. In these scenarios the system can model “next best route” according to resources that are left. This enables automatic rerouting and fail-safe routines when some parts of the assembly line are down. Putnik (2007, p. 1) also puts out the same benefits, as ubiquitous system has possibilities of increasing flexibility, reconfiguration and fault tolerance.

Apart from object and process automation, enhanced human-machine interaction allows better safety, when for example machines stop when human comes too near. As all or most of the objects are tracked, the safety function can work as a framework default, as in “if something is not where it is supposed to be, stop”. This way the safety function can also be used in greater extend, as a default. Of course automatic safety will not be totally foolproof, as machines might need some time for stopping and stopping is not always possible. But nevertheless it is a step into a safer direction.

Ubiquitous manufacturing can possibly have other positive effect on human resources as well. Monitoring the workers allows better, more even load planning for individual workers, and if the process can aid worker through instructions and helping in navigation, it can ease the load on workers memory and skills. Through this it is possible even to have effect on amount of mistakes and errors a worker does, which has direct impact into quality and number of defects. More timely use of human resources can also reduce the share of non-productive work, consequently increasing the share of productive work.

Besides for the plant working more efficiently, Putnik (2010, p. 1) reminds that a big part of the ubiquitous concept is to make the manufacturing line more controllable. This easier controllability can then be supplied to all relevant workers through handheld devices and mobile access to process controls. Better controls and interfaces combined with modular processes also lead to more flexible manufacturing and shorter production cycles (Zuehlke 2010, pp. 132-133).

4.4.1. Collecting data from the production line

Another advantage of transparency and visibility to shop floor is the possibility to collect data and record the actions directly from the production line. This recorded information can be then used in real time to further evaluate and evolve processes. The availability of this information has been regarded in various sources with various terms, such as real-world visibility (Haller et al. 2008, p. 3) or high resolution management (Subirana et al. 2006).

The collectible data could include for example process run times, material consumption, product manufacturing times phase to phase, the load and usage percentage of resources, idle times and time spent in queues, waste and loss of resources, number of defects and quality analysis of output. Also the generated meta-data enables traceability of the products, so that if some defects are found in the batch of products, it is possible to trace back to the origin of the problems. Saaksvuori & Immonen (2008, pp.117-199) also press the importance of traceability, and how it is enabled by being able to sort out the product into individual batches of components and software versions. The good traceability of products can be also seen as sensible risk management from production point of view (ibid.). A graph representing the information related to traceability can be seen at fig. 4.2.

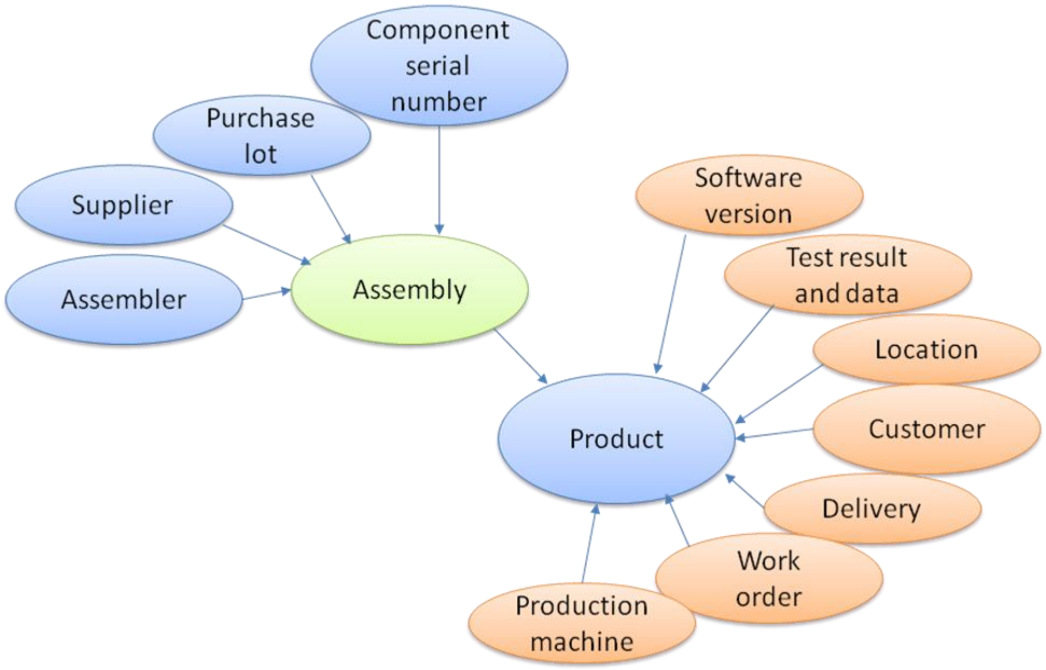


Figure 4.2: Information related to traceability of the object, adapted from Saaksvuori & Immonen (2008, pp. 119)

The increased accuracy and timeliness of data from manufacturing also grants deeper insights into processes, allowing better understanding and optimisation of them (Haller et al. 2008, p. 3). Another common example is the automatic shelf replenishing & procurement, and the accurate real world checks form a foundation for event-driven management (ibid.).

Meyer et al. (2009, pp. 22-23) also discusses that real-time data management enables developing early warning systems, that calculate trend values of production processes and pick up deviations that indicate something is not quite right. The information collected by the production can also be distributed to internal business intelligence (BI) for further evaluation, such as data mining (Meyer et al. 2009, pp. 149-150). This data can later be also used for better functionality in decision support systems.

4.4.2. Business process decomposition

Haller et al. (2008, p. 2) points out business process decomposition as another major paradigm for business value generation in ubiquitous computing. One of the main benefits from manufacturing point of view is that the manufacturing process can be split into very small units. The linkage to the PDM is also evident here, as these same small units are exactly the same that are needed to manufacture a product. This leads to possibility of modelling a product as a sum of process actions in addition to sum or structure of components. This aspect of PDM formation is further discussed in chapter 6.

Other advantages brought by business process decomposition can be increased scalability and performance of processes, leading to better decision making (Haller et al. 2008, pp. 3-5). This decomposition of processes also allows the process logic and “intelligence” to fall down closer to the actual implementation of logic. This further affects the responsiveness and scalability of the production processes. (ibid.)

4.5. Manufacturing Executive Systems, MES

MES are not directly linked with ubiquitous manufacturing, but rather are enabled by the information generated by ubiquitous systems. Meyer et al. (2009, pp. 1-2). The basic idea of MES is to add functions for planning, logging and control that act and react in real time (ibid.). Another term for MES discussed in (ibid.) is collaborative production management (CPM), which emphasizes the collaboration and information integration of the whole company. This information integration leads to the familiar concept of “digital factory” that we discussed earlier.

The main functionality of MES is to provide lower level “ERP” with production orientation for controlling manufacturing actions and processes (Meyer et al. 2009, pp. 19). Meyer et al. (2009. p. 85) also mentions order arranging from ERP system into optimal production sequence as core functionality of a MES. MES can also be seen to work as a virtual work queue, representing the digital product data interaction with the process data. So to sum up this means that the whole manufacturing flow from design and planning to execution is handled by MES (ibid.). MES can also be seen as a possible link between product data and work flow management, integrating “static” product data to operative data.

According to Meyer et al. (2009, pp. 11-12) the elements of an integrated MES are complete technical product description, resource management, planning and order management, performance monitoring, performance data recording and information management. Also in relevance to PLM, Meyer et al (2009, p. 13) promotes MES as the decisive integration platform for PLM functionality. Therefore MES can easily be seen as a central component in ubiquitous manufacturing systems as well, but there are not too many studies available on this topic yet.

The ubiquitous computing idea of the system being accessible through any computer interface can also be offered by MES. In this kind of implementation the MES works as the key entry point for human interaction, whereas the devices and processes can otherwise communicate more automatically without the consensus and constant awareness of the worker.

5. CRITICAL PDM FACTORS AND UBIQUITOUS MANUFACTURING REQUIREMENTS

In this chapter the requirements on PDM set by ubiquitous manufacturing are discussed. These requirements are based on the ubiquitous manufacturing concepts defined in chapter 4, and on the PDM factors defined earlier in chapter 3.

Kunar & Midha (2004, s. 58) propose the correct identification of PDM requirements and prioritizing the system capabilities as the single most important steps contributing to the successful implementation of a PDM system. This of course enforces the reality that one kind of PDM does not fit all companies, but in this chapter the best practices in PDM are explored. It is also given some thought whether or not these best practices are portable to large scale of various requirements and industrial areas.

Putnik et al. (2007, p. 3204) states model designing for product information, product management information, processes and resources as well as integration between design functions, and computer manufacturing control with production planning and control functions as the main activities for enabling ubiquitous manufacturing. The vision (ibid.) seems to sum up the requirements for product-process integration in ubiquitous manufacturing system.

Meyer et al. (2009, pp. 7-9) sums up the requirements of MES into standardization, data integration, engineering automation and process & change management. The data requirements set by a modern MES concept on the company go pretty much hand in hand with requirements of ubiquitous manufacturing, as they are both found on digital real time factory information.

5.1. The gap in product data between design and manufacturing

Lanz (2010, pp. 4-6) argues that the product data management should be consistent through all the phases from product design to process and assembly design. If all the data used is derived from single master data, there is no need for manual reforming and uploading of data, which is a time consuming and essentially obsolete task (ibid., Kropsu-Vehkaperä et al. 2009, p. 770). Besides being cumbersome and slow, there is always the risk of losing some data in translation and having to check to see if the product data matches in each department and phase of production.

To further increase the usefulness of product data, the data could be ultimately (in the future) formed in such semantic fashion that it is possible for a computer to deduct how to manufacture such item just by reading the product data. With such harmonized product data, the generation and evaluation of assembly lines could be greatly eased. Though this would also require that the assembly process could be described and modelled as functions, and could reconfigure itself to fulfil the needs set by each product.

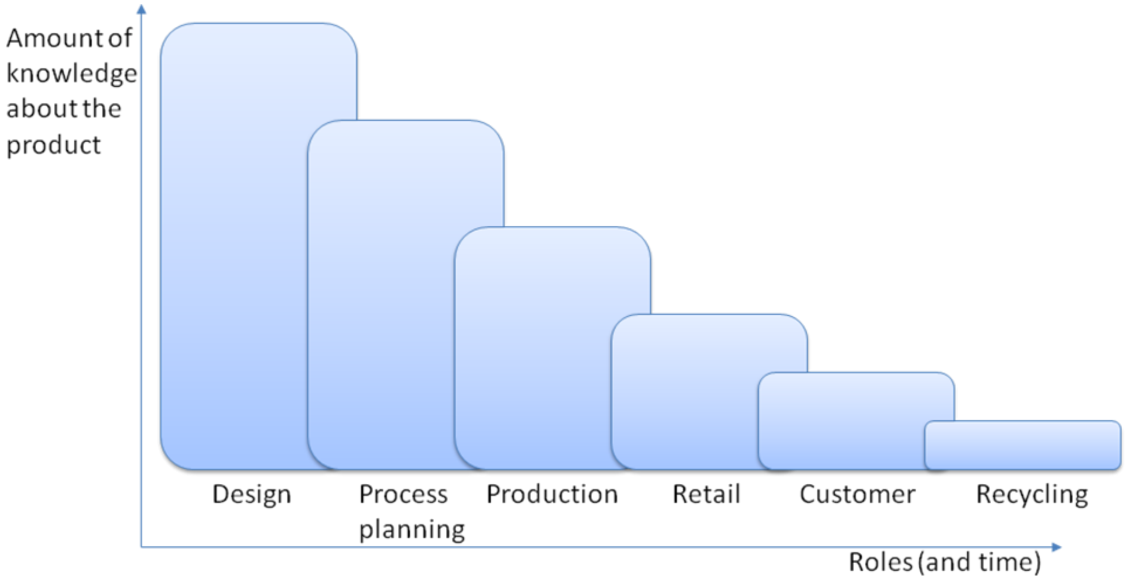


Figure 5.1: The decline of product data in relation to roles that process the actual product in chronological order. Modified from (Thomas et al. 1999, p. 54)

Fig. 5.1 illustrates the generalization that the information about the product diminishes through every phase on its lifecycle. The knowledge does not degrade over time, in the contrary the knowledge about the product for a single role is most likely to increase over time in a single role. For an example the design could be revised after initial versions, production processes enhanced and even customer is more likely to know more about the product he or she is using the more it is used.

This is rather one sided view on the knowledge, as it does not take into account the actual need of information for each of these phases. A customer most likely will not be too interested about how the production systems work or how the manufacturing for the products was planned. Another thing this figure does not take into account is the role specific knowledge that is not directly derived from the preceding role. This means knowledge that is not used or generated by the preceding phase, such as worker timetables in production.

But the question that the fig. 5.1 clearly does bring to the table is that should there be such declining of knowledge, and would it enhance the following roles if there was no such decline. Lanz (2010, p. 4) also reminds that every time the product related documentation is transformed to different format, it will lose some information because of the different systems and formats used in different phases of product lifecycle, which is one of the reasons for this diminishing of product related information.

5.2. Linking process information into PDM systems

To add ubiquitous elements into PDM system, it is crucial that the process information can be accessed as well. If only embedding the product information into products themselves, the rewards for ubiquitous manufacturing are quite limited. But taking processes into consideration opens up more seamless product – process interaction. As a single product is the summary of multiple processes, a product could be defined as a sum of certain processes and components. This makes it possible to add the manufacturing steps into the component models discussed earlier in chapter 2, and an example product structure that is expanded to contain process information can be found in chapter 6.2.

The readiness to support ubiquitous manufacturing is clearly not the only motivation to harmonize product and process related databases of the enterprise. Lanz (2010) covers quite thoroughly these issues, and they are discussed further in chapter 5.4.

In order for the factory to work digitally and automated, the processes have to be digitalized as well. Lanz (2010, p. 27) also mentions that digitalized processes also enable automated assembly process planning. But in the sense of PDM and in the scope of this study, the focus is only in linking the process information into PDM. Of course this works as a foundation for automated processes, to already have PDM handle the process instructions for producing the items in question

5.3. Foundation of ubiquitous functionality in company IT

This chapter discusses whether PDM or ERP system should work as a foundation for building the functionality to support ubiquitous manufacturing.

Reasonably it can be also argued whether the processes should be linked into PDM system, or the PDM system more integrated into ERP system. In this study the PDM is chosen as a starting point, as components and products are easier to model as object oriented databases. Also the definitions of processes can be rather vague compared to definite components and products. Surely we can argue if ERP holds the descriptions of processes either, as the role of ERP can vary greatly from company to company due to various modules it may or may not contain. Also the whole definition of ERP can be currently seen as quite vague with all the earlier iterations of ERP such as material requirement planning (MRP) and material resource planning (MRP II) are more or less fused into one term.

Another point for PDM headed implementation of ubiquitous manufacturing is that product data is more granular and possibly easier to digitalize than process data. Also the product data does not usually contain functionality, which further simplifies the interface into simple attribute setters and getters, whereas process interface might need some much more sophisticated functionality.

The product databases already collect big amounts of information that can be used for the manufacturing, such as blueprints and CAD schemas. To include the exact processing information and such into PDM seems only appropriate. Lanz (2010, p. 1-3) also reminds that the processes and assembly are more downstream actions than product design, as the processes are usually defined later on so that they produce the product as designed. This idea makes product design and product data a natural “starting point” situated upstream for the system renewal.

Meyer et al. (2009, pp. 1-2, 8-9) also criticises traditional ERP systems about being largely focused only on administrative and accounting systems, whereas the focus for production processes and products would be more valuable in manufacturing environment. The view by (ibid.) then again focuses on implementing a MES system for manufacturing, situated somewhere more downstream from ERP, closer to production. But Meyer et al. (ibid.) also visions MES as a “system of systems”, that governs other manufacturing systems alike. In situation like this the database still needs to be formed, and product data seems a viable place to situate ubiquitous manufacturing data.

5.4. Formalized knowledge transfer

Another idea that heavily enables ubiquitous manufacturing is formalized and harmonized knowledge transfer between product design, process planning and production. Lanz (2010, pp. 3-6) reminds that if data is manually transferred and transformed, besides being slow and prone to error manual work, some data is almost always lost. Even if no actual data is lost, the metadata, interoperability and traceability of the data can possibly suffer. The data transfer is usually also one way progress that does not per default support interactive communication, such as sending the data back to revision with markings. Another problem closely related to this is that the changes done to product data “upstream” are not automatically reflected downstream, and when they are they make current documents obsolete, and the transfer process has to be done again. (ibid.)

Besides the manual transformation in different phases of product design and process design as discussed earlier in chapter 5.1, the master data model could also cover the process input. The formalized knowledge transfer aims to eliminate the need for these manual transformations, as the data should be consistently transferred through the whole manufacturing cycle. Of course various phases of the product development need different models and data, but all of those should be automatically derived from the master model, instead of being transformed, frozen views or snapshots of the model.

Meyer et al. (2009 p. 30) discuss that advanced production systems such as MES require a complete and consistent data model that contains the production and resource mapping as well as product definition data. Meyer et al. (ibid.) also propose that the solution for the fragmented data could be master data management, or master data model. This would make it possible that processes and data is situated at only one place in the company’s IT system (Meyer et al. 2009, pp. 129-130).

5.4.1. Capability to process real data

For the process and product systems to communicate, one of the first things needed for ubiquitous manufacturing is the access to the real world data, such as actual locations and states of objects. The trinity of these data models is shown in fig. 5.2, and this can be seen as the core concept of product data management in ubiquitous manufacturing. Linking process information into product information, and linking them both to concrete, real information of both processes and products, so that all information is interconnected.

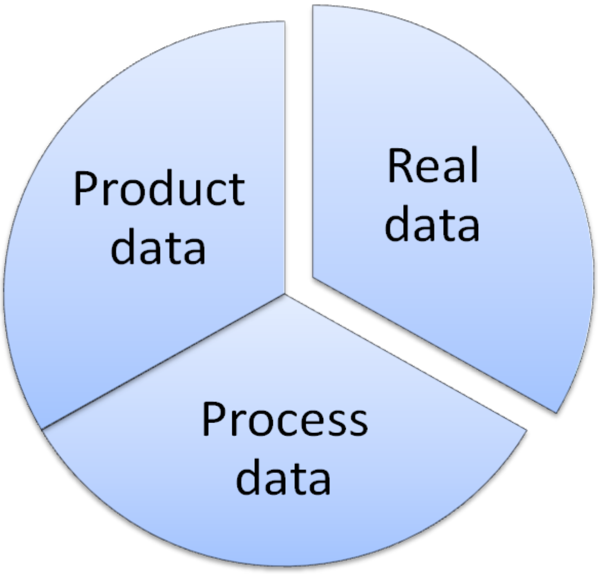


Figure 5.2: Relationship requirements of data in PDM

The real or concrete data is still a quite tricky term to use, as the information related to products can be real in either collective sense, or in a singular sense (regarding a single item). To visualize this problem, see fig. 5.3, where the product data is viewed in a table with axes of collectiveness and concreteness.

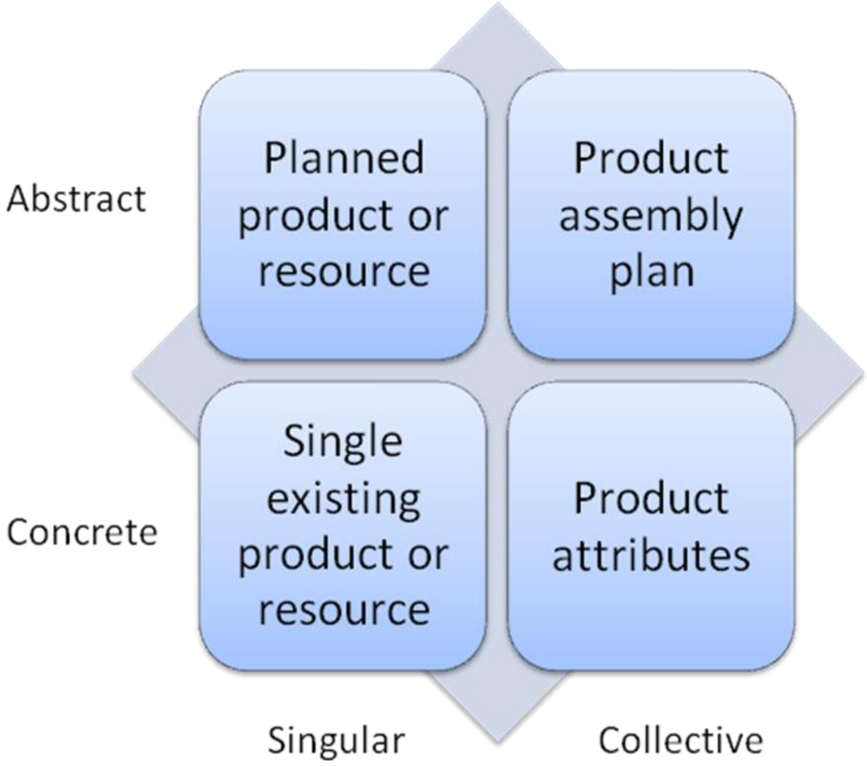


Figure 5.3: Product data examples in terms of singularity and actuality

Abstract singular product data is something that defines a single object, item or resource, which does not yet exist or is not yet brought to use. A singular concrete item is an already existing single object. Collective abstract product data would describe something non existent that regards all the products of certain type, such as an assembly plan for products of certain type. Concrete collective data means something that defines all the actual real world objects of certain type, such as product attributes like size and specifications.

With these figures (5.2, 5.3) it is easy to see that the generic PDM focus is on collective attributes of the objects, whereas ubiquitous manufacturing shifts the pressure to include singular objects as well. This division of product data is further used in determining the data models required in ubiquitous manufacturing in chapter 6.

Therefore singular level item management can be formulated as a ubiquitous manufacturing requirement, because that is the only way to bind item specific real world data into objects of production line.

5.5. Ontologies and semantic data

Ontology in computer and information science is commonly defined as an explicit specification of shared conceptualization (Brunner et al. 2007, p. 748). As conceptualization means the modelling of objects and concepts and their relationships that exist in certain area of interest (Gruber 1993, p.1-2). Gruber (1993, p. 5) also clarifies this by definition that conceptual schemas define relations in data, but that ontology define relations in higher level knowledge instead.

Shadbolt et al. (2006, p. 100) define ontologies as means to make an explicit commitment to shared meaning among an interested community, but while anyone can use them to describe their own data.

Currently ontologies are claimed to be one of the key drivers in data integration and generation of common conceptualization (Shadbolt et al. 2006, pp. 96-97). Sciences in general have had a need to understand systems across wide ranges of scale and distribution, and ontologies are seen as a possible remedy for that (ibid.). This can be seen as a clear analogy to manufacturing context, where the information about product and its manufacturing is also greatly distributed.

5.5.1. Web Ontology Language

Web Ontology Language (OWL) is a collection of knowledge modelling languages, backed by the World Wide Web Consortium (W3C). According to its standardiser W3C (2010), the main idea of OWL is to create more machine-readable documents, that automation and applications can easily read and follow, apart from being readable just by humans.

OWL consists of three sublanguages of varying level of detail. The simplest of these three, OWL Lite, includes classification hierarchy and simple constraints. It can be used to build hierarchies of elements with only truth values (0 or 1, yes or no). The more advanced, appropriately named OWL DL (Description Logic) also includes description logics and computational completeness in addition to the functionality of OWL Lite. The most complex version of OWL languages is OWL Full, which crops out the reasoning support present in simpler OWL versions, but includes syntactic freedom. (WC3, 2010)

Brunner et al. (2007, p. 749) criticizes OWL Lite for lacking the functionalities required by product information management (PIM), and OWL Full for possible computational difficulties because of syntactic freedom. Thus OWL DL being best suited for product information and product data management (ibid.).

5.5.2. Machine-readable, semantic documents

Semantic web, that is currently also titled as web 3.0, has also received quite a bit of interest recently. The idea of the semantic web is to form websites so that they are also machine / computer -readable, instead of being just made for humans (Shadbolt et al. 2006, p. 96). This means that the computer can use generic logic and deducting to find the required information from the document (ibid.). The same analogy can yet again also be seen in product data, so that it should be made machine-readable instead of being made just for human viewers. Ultimately if the product is generally made by machines, making product documentary machine-readable in the first place would seem rather sensible. Even more so, in a manufacturing system that is planned to work ubiquitously and automatically, where the components of the process communicate with each other.

Single commonly used mechanism for semantic web is Resource Description Framework (RDF), or RDF Schema (RDFS) (Shadbolt et al. 2006, 98). RDF uses very simple logic of all object relations being modelled in data triples, so that there is always a subject, a predicate and an object (ibid, Kortelainen & Mikkola 2010, pp. 343-344). The power of RDF being that it is very natural to read for humans as well. Example of this kind of data triple could simply be “John is a person”. Kortelainen & Mikkola (2010, pp. 350-351) define these semantic data structures as flexible, expressive and simple way to integrate all product related data. An illustration about the relationships of the RDF, XML, and OWL technologies in semantic web context can be found in appendix B. The illustration shows the hierarchy of these technologies, and how the higher level technologies are based on services offered by lower level technologies.

Some criticism towards semantics and triple store ontologies is that there are currently only few industrial solutions available, and that managing them requires lots of computational resources (Brunner et al. 2007, pp. 753-754).

5.6. Performance requirements

Besides setting lots of requirements for the contents and architecture of the database, the PDM system is under strict requirements concerning the performance of the database system. This boils down to the fact that the PDM database must be able to process this enhanced amount of data, but also vast amount of queries generated by multiple linked objects and processes of the system. Meyer et al. (2009, pp. 13-15) also emphasize the requirement for real-time data management, as excessive response times can impair the whole production system. Besides impairing the production, if the real-time data is not available, the system is not able to supply any of the ubiquitous functionality that requires real-time data either, as discussed earlier in chapter 4.4.

Industrial level applications can be very demanding on databases, requiring thousands or possibly even millions of operations from large databases every minute (Brunner et al. 2007, p. 753). One can easily vision that monitoring every single object of meaningful size / function in a manufacturing site is requires vast amounts of processing. But as already said in chapter 2.2, the constant advancement in computer technology and in CPU speeds enables new solutions every day, so the technology and hardware side can be presumed to catch up sooner or later, while the ideology remains the same.

Another possible way to combat the performance requirements on the system is to distribute the logic and processing to objects. These thick clients would do most of it by themselves as discussed in chapter 4.2, leaving the central computer / database resources available for higher level tasks.

6. PDM FORMATION IN UBIQUITOUS MANUFACTURING ENVIRONMENT

This chapter discusses the means of fulfilling the requirements set to PDM systems by ubiquitous manufacturing, which were posed in chapter 5. First the completion of the empirical study is briefly defined, followed by architecture and hierarchy aspects of the model are described. Finally in the chapter 6.4 a product structure model is created for the ubiquitous manufacturing project.

The ubiquitous manufacturing concepts are heavily information oriented, and the whole functionality of the system leans much into having more information available than the current product models and PDM systems have. Because of the great need for information, the product structure model for the system is a crucial part, functioning as a “heart” of the system. All the logic and functionality that further enables the ambient and ubiquitous intelligence are later built to exploit the stored data. The model defined here is further used in realizing a prototype production line in chapter 7.

Besides product structure, the formation also commits upon title management, as the product structure elements are defined. The document and version management aspects of PDM are left for lesser study, as they can be seen to function in more mundane, supporting tasks for the system. The document management with ontologies is discussed in chapter 6.6, and version management briefly in chapter 6.5.

6.1. Execution of the empirical research

The model represented here was modelled after the requirement set posed in the literature review performed in the theory section of the study. The requirements set by ubiquitous computing were fulfilled in the sense of accuracy that was possible for the scope of the study.

The requirements posed for the model earlier in the study can be summarized as:

- Identifiable single objects and process activities
- Single item level data management
- Process and activity mapping to PDM
- Real data management capability (such as location, state and time)
- Real time data management capability (performance)
- Formalized data transfer through the lifecycle of product (birth of life)
- Historical data collection for data analysis / mining
- Availability and accessibility of data

The product structure model is only a data model, which also means that it does not take into account the functional requirements, such as performance or accessibility requirements. Clearly the data model by itself does not perform any ubiquitous functionality, but works as a foundation a ubiquitous manufacturing system can be built upon. The prototype product line in chapter 7 is a single realized ubiquitous manufacturing concept, enabled by this data model and other technologies that are used there. During the study the pilot product line and the data model were developed simultaneously, both helping to define each other by means of functionality and requirements.

6.2. Generic architecture

Because of the requirement of singular object level data management, all the physical objects need to have an equivalent in the database for all the instanced data. This adds a need to split the database into real world layer and abstract layer. The collective, abstract layer for a product contains all the product related data that is not instanced, that all the products of the same model share. Whereas instanced, real data naturally contains only singular data of single objects, such as ID, location, customization flags and temporal data like state. See fig. 6.1 for simple UML model representing this. The fig. 6.1 depicts the collective classes such as components or resources can have multiple singular real world instances, whereas these instances only relate to the collective class they are instanced from.

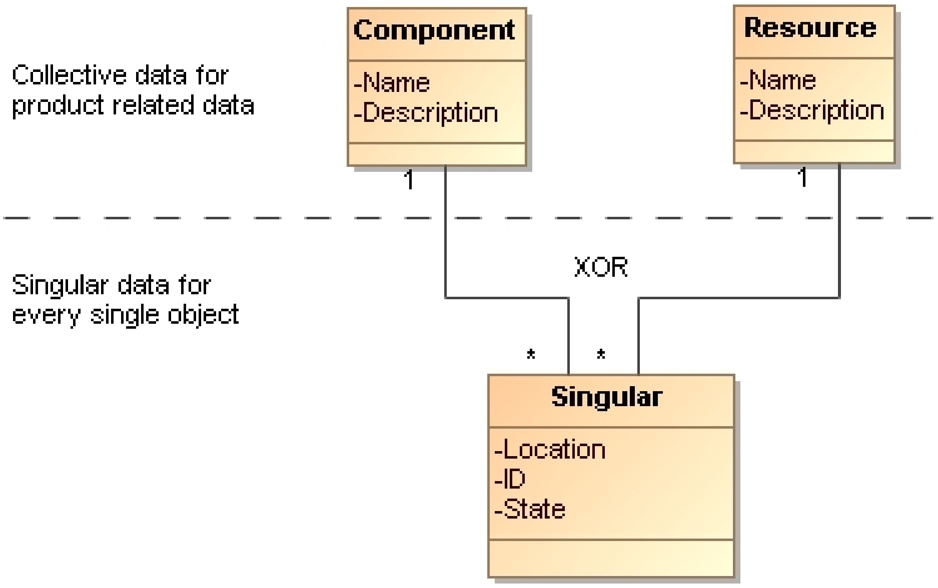


Figure 6.1: UML model representing the relationship between collective and singular data

This way it is clear which parts of the database represent concrete real-world objects and which form the collective, more classical view of products. The navigation of the product structures in the database also stays sensible if all the navigation is done through collective phase, and when needed it is possible to drill down to concrete singular level. This possibility to drill down is the very same here as in the high resolution management concept discussed earlier in chapter 4.4.1.

6.3. Recursive hierarchy

In order to satisfy the needs of product-process interaction discussed in chapter 5.2, the manufacturing processes and activities also need to have their digital equivalents in the system. This is done so that a product can be modelled as a sum of processes and actions, besides just being modelled as a manufacturing bill of materials (MBOM) or list of components. These manufacturing activities can then be modelled as requirement lists, so that they define what resources or components they need to be done. This also works as the connection to PDM, as the product can be then defined as modelled activities.

This is an important step in ubiquitous manufacturing PDM, as binding production processes and phases into products and their components is a requisite for digital factory, digitalizing the production line even more. Of course the processes and activities need to be added into database as well, both in abstract and concrete, singular items, as discussed in chapter 6.1. These singular activities can be then used for example as production plans or orders, such as “paint three car bodies”.

When all the appropriate processes, tools and phases of the production are in the database, the whole manufacturing of a product can be modelled with hierarchical classes. In earlier chapter there was discussion about representing product structure as a hierarchical tree that contains all the components, but this tree form can be expanded to include even the processes. This way for an example the single product manufacturing is divided into phases, and phases into events. Then all the resources needed to complete the event are linked into the event, and summing up events and phases formulates the whole requirement set for the manufacturing of a single item. A resource in this case means anything needed to assemble a product, including parts and components but also machines, tools and workers (or work). See fig. 6.2 for an example of this kind of hierarchical product model.

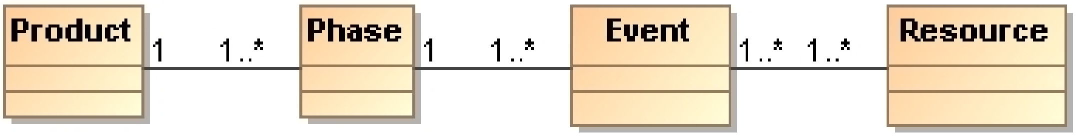


Figure 6.2: The hierarchy of product definition

The linking between product components and manufacturing processes can also include information and instructions on how to complete the manufacturing step, such as the correct machine instructions and settings for operating the component. When the hierarchy is presented like in fig. 6.2, a single level of manufacturing is shown. If needed, the hierarchy can also reach lower levels. In cases where the resources used can be subassemblies, and as such they are also modelled as products, so that they have subcomponents and manufacturing steps as well. This can be done recursively, so that it is possible to drill in as deep as the database contains subcomponents.

6.4. Database model for product structure

In this chapter the database model used in the project is defined. The central ubiquitous manufacturing requirements and their implementation to the data model are discussed here.

6.4.1. Using the Product-Process-System model as a basis

The product-process-system (PPS) model of Lanz (2010, p. 90) aims to cover largely the same field of integrating processes to product data, and also to system data. See the PPS-model in appendix C.

The PPS –model is based on Core Product Model (CPM) and Open Assembly Model (OAM) which were both designed by National Institute of Standards and Technology

(NIST) (Lanz 2010, pp. 25-27). CPM aims to model attributes of a single product, whereas OAM covers the assembly of a product. The PPS –model itself focuses on modeling the product and actions that occur upon it in manufacturing (Lanz 2010, pp. 63-63, Sudarsan et al. 2005, pp. 1402-1405).

As the PPS model represents well some of the areas needed in this study, the model is used as a foundation for modelling the database model for the project. Naturally PPS model does not take into account most of the ubiquitous functionality, such as singular, real world representations of objects nor the real data, which are consequently the biggest modifications done into it. The naming of entities on the PPS model is rather unambiguous, so it is adapted as well. Also the scope of Lanz's model (ibid.) is too accurate for this study, which leads to some parts of the PPS model are cropped from the model. The cropped parts are mainly the precise physical and geometry representations of objects, the high depth of system modelling and the reasoning engine functionality.

Naturally such things as the precise geometry data etc. would be very helpful to have in a product model, but they are left out as the scope of this study is limited. The precise geometry data would be one of the main elements for creating a totally semantic product structure model that could allow self-configuring processes for automated assembly lines, as discussed in chapter 4.

6.4.2. Binding processes to product structure

In order to be able to model the assembly, the phases and events of manufacturing a product have to be tied into the product, along with the list of parts and components. This way it is possible to determine the assembly process phase to phase, and see the component / part requirements for each phase separately. The product structure shown in fig. 6.2 handles the simple connections between processing phases and events related to products and resources.

The PPS model of Lanz (2010, p. 90) models processes split into smaller pieces, that are tasks, operations, actions and sub actions. As for the scope of this study, the two smallest units, actions and sub actions are cropped, but they could be easily re-implemented if needed. So the hierarchy is such that multiple operations form a single task, and multiple tasks form a single process. A final product itself consists only of multiple processes. This binding between processes and products at least partially answers the requirements for bridging the gap between product data in design and in manufacturing.

6.4.3. Binding real world data to PPS model

Apart from containing processes / activities and the product structure, ubiquitous manufacturing adds the requirement of “real data”, that is correct digitalized information about a single object or entity in the manufacturing. This single entity can be a part or a component, a final product, an event or activity or a production resource such as a machine or a worker. The real data needed to bind into this entity includes state, location, possible orders and queues and other flags. The data entity containing the real data also needs to have a pointer or a link to its collective counterpart as well, so that it is possible to determine the non variable attributes of the entity. These collective attributes include the classic product data, that all the entities of a certain product type share.

The simple binding between singular objects and collective attributes is already discussed in chapter 6.1, but there still remains the need for instancing production orders and queues.

Also even though the collective data models are instanced as singular data, there still lies a need to construct the singular item model from all the singular items attached into it. The need for having these two different models (collective product model and singular product model) ranges from the fact that all the collective data should not be stored in all the singular objects, as it is a waste of disk space and database efficiency.

6.4.4. Ubiquitous manufacturing product model

The database model is based on Lanz’s PPS model and the ubiquitous manufacturing requirements defined in chapter 5. For the UML class chart of the model, see fig. 6.3. There is also SQL compliant version of the model in appendix D.

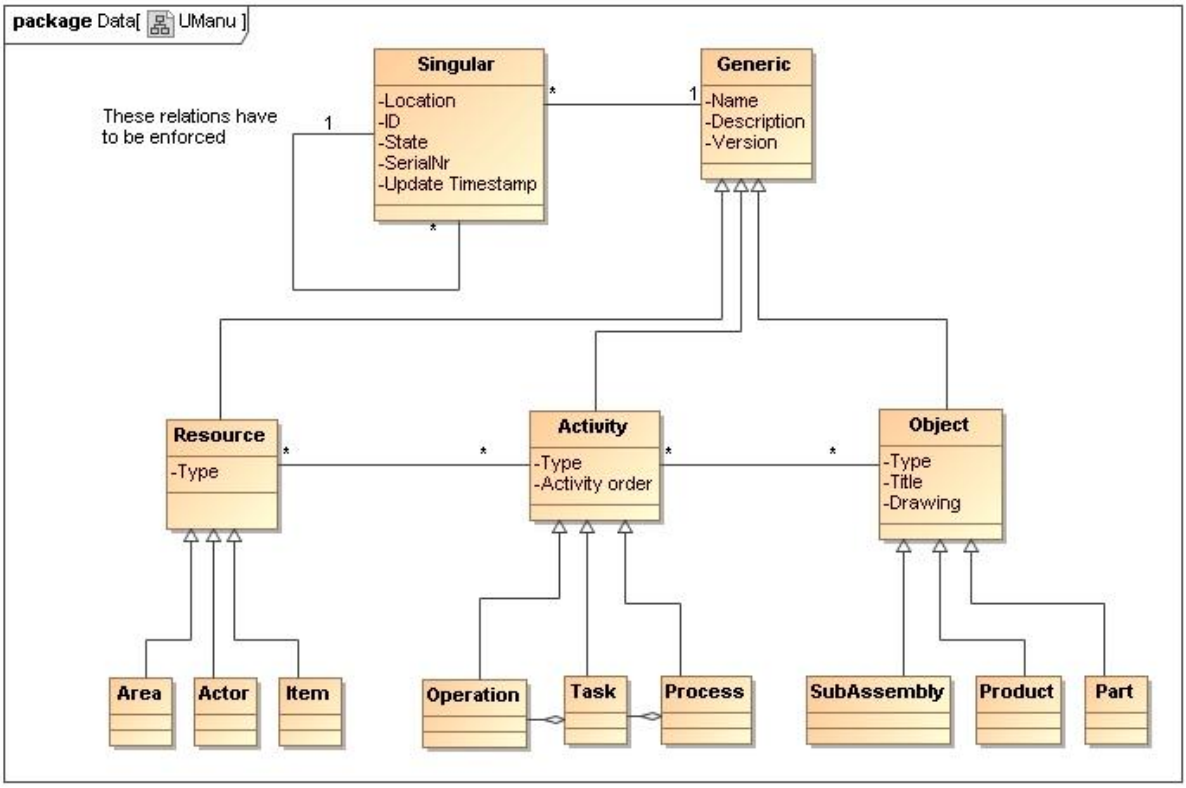


Figure 6.3: The ubiquitous manufacturing product model

The main features of the model are the ability to model a single product in terms of activities and resources, and the singular structure for modelling the concrete real-world objects, resources and activities.

The model follows Lanz's PPS model in categorizing the entities of a production line into resources, activities and objects. These three classes act as the main parent classes of each genre, and are further inherited by their subclasses. All of these three main classes inherit the most general attributes from Generic class which is an abstract, purely virtual class that represents the generic, non singular, nature of the classes.

The Generic class contains only basic information of the entity, such as name and description which can be found on every entity. Also version of the entity is situated here for simple version management. However the main function of the Generic class is to work as a bridge to Singular class, so that any subclass of Generic can have a Singular aspect as well.

The Singular class encapsulates the binding to real world, containing the component identification (serial number for example), RFID or UWB –tag ID, state and physical location. The serial number of the product and its tag ID could also be the same thing, in cases where the tag follows the object through its whole lifecycle. But in case of transferrable and reusable tags, there is a need to separate tag ID and possible component serial number. The Singular class also contains connections to other Singular entities to model the real relationship between real objects. This connection could also be modelled as rigidly as the whole generic item structure (into relations of objects to activities and activities to resources), but is left open in this implementation. To rigidly model the object-activity-resource relations would rather straightforwardly mean building the same architecture present in generic entities to Singular entities as well. But this would also increase the complexity and more importantly the amount of redundant data. In this case the legality of singular relations to each other can be enforced through checks from the Generic entity bound into the Singular entity.

The Resource class is inherited by various different types of resources that are used / needed to produce objects. In this case resource is used to name entities that can be used multiple times to produce objects, and are not consumed by usage. This is the main difference between resource and a component of a product. Types of resources include areas (facilities and their attributes), actors (workers, robots etc.) and items such as tools.

The Activity class subclasses form the hierarchy of actions in a product assembly. The highest level of activity is a process that contains multiple tasks. These tasks can then again subsequently contain multiple operations. Each of these operations contains links to the resources and objects needed to successfully complete them, and then tasks and processes can combine these into larger, complete requirement lists for the production. For an example a process could mean a single larger part of manufacturing a product, such as painting & coating, where tasks could be applying primer, applying paint and applying varnish. Operations in this case would be instructions on how to apply these chemicals on the surface.

The Object class works as a parent for all product structure (or BOM) related entities. Subclasses of Object are Product, Part and Sub Assembly, which are rather self explanatory. Subassembly itself is can be modelled as processes, describing the structure of the subassembly in activities in similar way that the product is described. This allows the recursive hierarchy discussed in chapter 6.3. Product is the highest level entity, such as final goods ready to be shipped.

See fig. 6.4 for opened version of the class diagram showing the relations as a tree structure. In the fig. 6.4 the tree structure only shows three processes on product etc, but the number of these sub phases can be anything required. Also an operation can have any number of resources of any type (for example multiple items and actors) instead of just one of each as portrayed in the fig. 6.4

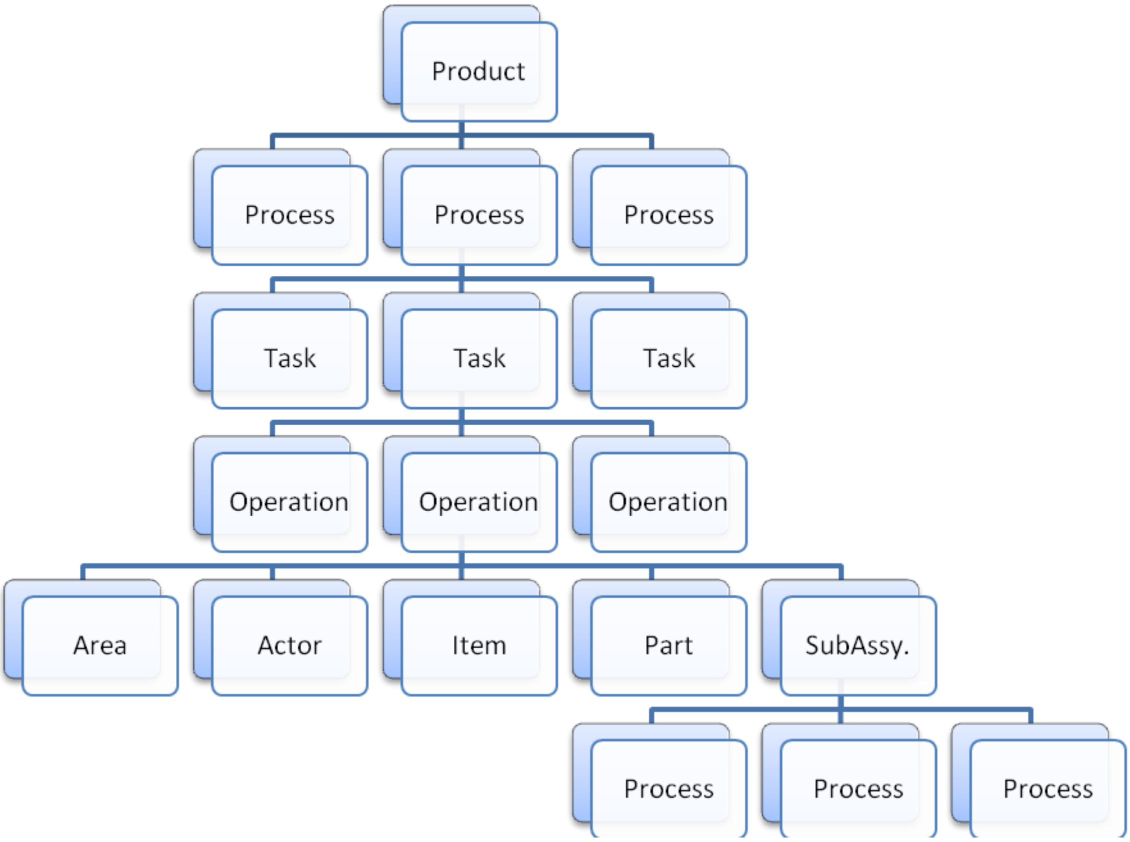


Figure 6.4: Example of the data model as a tree structure (only ones item per level is expanded for clarity)

6.4.5. Example product modelled with the UM product model

To further illustrate the ubiquitous manufacturing model product defined in this chapter, here is an example of a simple product modelled with the UM product model. See appendix E for the model of the same ballpoint pen used as an example earlier in chapter 3. The model is now modelled with depth of only one activity per level, as the ballpoint pen example is trivial in that sense, and only assembly is considered. This is partially because the example only ranges to assembly from already made parts, and does not take into account the manufacturing of the parts. In cases where the model is used to manufacture the parts, the need for more activity levels becomes more evident.

The depth of the model gets very deep even in such a trivial model, because all the phases of the assembly have to be modelled as separate subassemblies.

6.5. Logging data for analysis and version management

The database designed in this chapter is first and foremost for transactions processing (OLTP), meaning that its main purpose is to “keep the wheels of the plant running”. For OLAP functionality, the historical data has to be stored as frozen states of the database during recorded time periods. This historical data and metadata generated by the system can be used for example in data mining and flow analysis, in order to gain insight into the production.

This analytical (OLAP) functionality requires adding the time dimension into the database. Generally PDM databases do not have this function, as the product descriptions are not supposed to vary over time, but rather over version history, if at all. The product descriptions do not change in the ubiquitous manufacturing model either, as only the singular objects change through time, because they contain instanced data, such as state and location. Therefore to complete logging, only the singular objects used in the data model (as described in chapter 6.4) should be stored with timestamps. The time stamped objects could then later be used in order to analyse the effectiveness of the plant. This digital monitoring makes it very easy to measure the manufacturing processes, as the measuring is constant and does not have to be set up just to measure. Although as the amount of data generated by this logging can quickly become monumental, there is a need for a sensible policy in determining the depth and accuracy of the tracing and logging.

For version management, every entity contains a version variable, which makes it possible to distinguish same generic type objects of various versions from each other. Also the bond between versions of same generic entity is present, as they both can have the same name, while indexing with name and version allows viewing all versions of chosen entity. Containing the version in Generic class makes it possible to use version control in activities and resources too, in addition to only using it in product and component related object class.

6.6. Document management with ubiquitous manufacturing product model

Besides unifying terminology, hierarchies and concepts of product management, the semantic ontologies also offer possibilities for database creation and document management via triple store databases. The semantic databases themselves do not store “documents”, but can contain pointers or resource identifiers (URIs) locating conventional documents, such as CAD files or brochures. Another notion to the document management is that the semantic database could possibly solve some problems regarding the supplying of different views to product data for different sections of organization. The same semantic data should be accessed by all parties that need the product data, but through different views and rights depending on the needs of the data user. This would ease the pressure of everyone making their own cropped versions of the same data, and in general would lessen the product information decline described in chapter 5.1.

Another closely related concept to document management is the concept of master data management. Loshin (2009, pp. 5-7) defines master data management as an integrated view on key business data that is replicated in multiple systems. Master data in scope of this study means that the product data being only a single entity that is referenced elsewhere instead of being duplicated in multiple different systems. Thus the master data management concept is not further explored. The document management aspect of the model was partially overlooked during the project because of its supporting functionality, as the focus of the study was on technologies and features of the data model that enable novelty functionality.

7. CASE: USING UBIQUITOUS TECHNOLOGY TO AID MANUAL MANUFACTURING

This chapter describes the “UManu” ubiquitous manufacturing project of VTT that was done alongside the study, and which the study was consequently part of. The chapter includes general description of the project and its concept, environment and technologies. As it also works as empirical part of the study, the chapter and the project relates to earlier parts of the study. In short the project aims to realize some of the ubiquitous manufacturing concepts defined in chapter 4, using the PDM data model defined in chapter 6.

7.1. Concept briefly

The prime idea of the project was to create a laboratory space where manufacturing is aided by computerized, digital production information flow. The simple production done in the laboratory is controlled by manufacturing execution system (MES). The production flow monitoring is implemented by tracing components and tools. The demonstration products that are assembled are modelled with ubiquitous manufacturing data model defined in chapter 6, and the same model is used to instruct and monitor the actual physical assembly of the product. Aiding the worker and instructions are done in two phases. These phases are helping the worker to find all the required components and tools for the production (kitting), and showing dynamic instructions to the worker about his current task in assembly phase. While aiding the worker, the same data of the production is used in MES to show the exact state and progress of the assembly.

The main objectives of the pilot production line are to monitor, control and plan assembly line operations through digital data. The data generated through location services and singular level object management works to bridge and link real world entities and their digital representations in the system.

7.2. Environment

The concept environment is a small to medium sized (SME) manufacturing company, where work is mostly done as computer assisted manual work. Of course some of the implications can be further assumed to be present in other environments as well, and some concepts for various environments are given thought. But the main focus is on aforementioned SME manufacturing.

The project itself is realized in VTT’s manufacturing laboratory in Otaniemi, Finland. The same physical space is also used for various other VTT manufacturing projects, some of which also work as “customers” for the ubiquitous manufacturing system. This is done so that the ubiquitous system serves the other project processes as MES functionality. The laboratory environment also makes it possible to materialize the concept to test and see how it works. The test environment naturally can not accurately simulate a real manufacturing shop floor, and evaluate the concrete effects on a manufacturing and throughput / lead times and such. But what it allows is to reiterate and enhance the concept of ubiquitous manufacturing, while allowing to find some problems that might surface for example from bad design.

A visualization of the manufacturing laboratory can be found in appendix F.

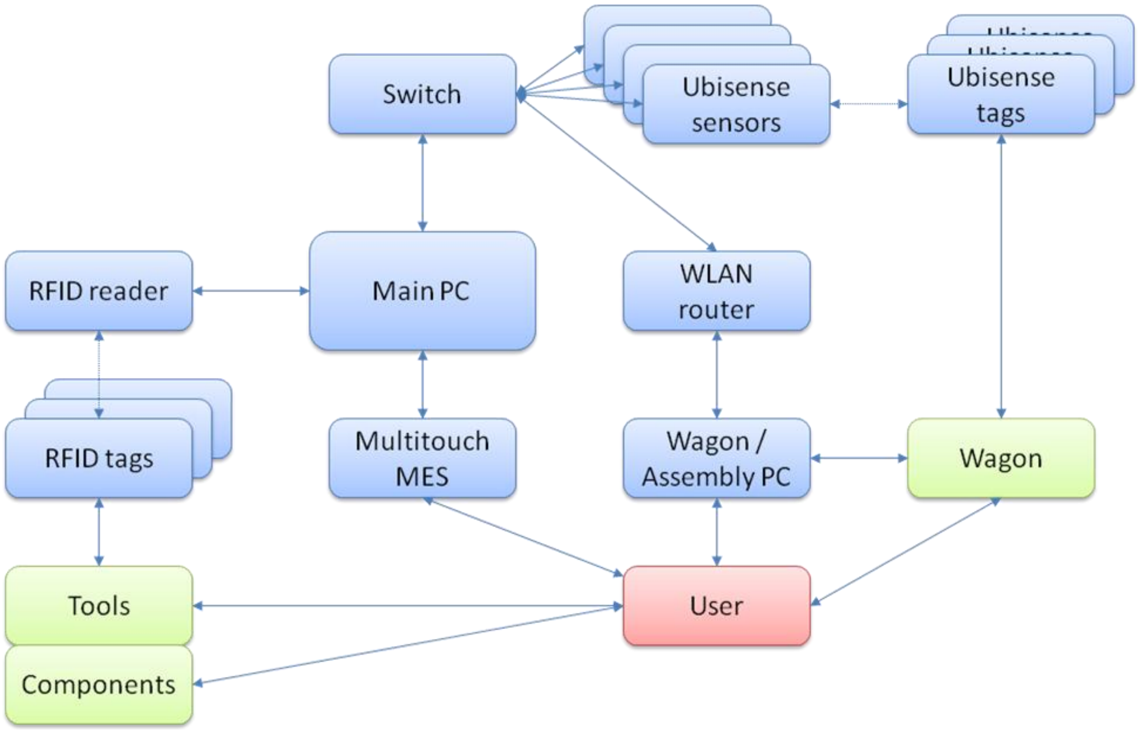


Figure 7.1: Devices, objects and connections in the production lab

Fig. 7.1 depicts the connection relations of the devices and actors used in the ubiquitous manufacturing pilot product line. The main PC is connected to MultiTouch MES and

RFID reader which reads the RFID tags contained in tools. Main PC is connected to Ubisense sensors through standard Ethernet LAN, and to the wagon PC through WLAN. Users of the system interact directly with tools, components, wagon, wagon PC and MES.

7.3. Technologies of the project

This chapter describes the technologies used in order to realize the concept described earlier. The real time location system is based on Ubisense network, close range locating is done with RFID and the user interface to MES is done with a MultiTouch display.

7.3.1. Ubisense RTLS network

The location services and object tracking in the laboratory are done via Ubisense real time location system (RTLS) network, which is a commercial product by Ubisense. The network consists of tags that are attached to tracked objects, and multiple sensors that are connected to the central computer executing the Ubisense location engine software.

The locating is done by ultra wide band (UWB) radio signals that are sent by the tags, and then received by the sensors. The sensor boxes have multiple antennae inside, so that the sensor can estimate the angle where the received signal originates from, and the distance to the signal source by the strength of the signal. This means that crude level locating can be done with just one sensor and a tag, but better results and accuracy are naturally gained by adding more sensors, so that they can compare their results via triangulation. With multiple sensors that are all connected to master sensor, it is possible also to use time difference of signal arrival to calculate the location of the tracked tag. The equipment manufacturer promotes that the equipment is capable of 30 cm locating accuracy, and the system that was built in laboratory space covered some areas with accuracy close to +/-20 cm. The area covered by the four antennae is roughly the whole manufacturing lab floor, although the locating accuracy drops drastically on the edges where all sensors cannot see the tag.

The tags themselves are battery operated, and their dimensions are 4cm (width) 4cm (length) and 1cm (depth). Their size limits their usability, as they are too big to be put on tools or smaller parts of components. However they can be used for tracking people, movable resources such as kitting wagons or movable robots or larger parts of the product, such as car bodies. Also as discussed earlier in chapter 4.3, there is no need to have UWB tags on objects that do not move freely inside the plant.

The location engine and sensors are calibrated so that the software controlling the sensors knows the exact locations of the sensors. When the sensor positions are fixed, the location software can deduct the location of the tags travelling in the monitored area.

A visualization of this traced area is shown in fig. 7.2, where the sensors are shown as corners of the quadrilateral, and the effective tracing area in grey. The tracing works outside the quadrant as well, but for precise location a tag has to be seen by at least two sensors simultaneously.

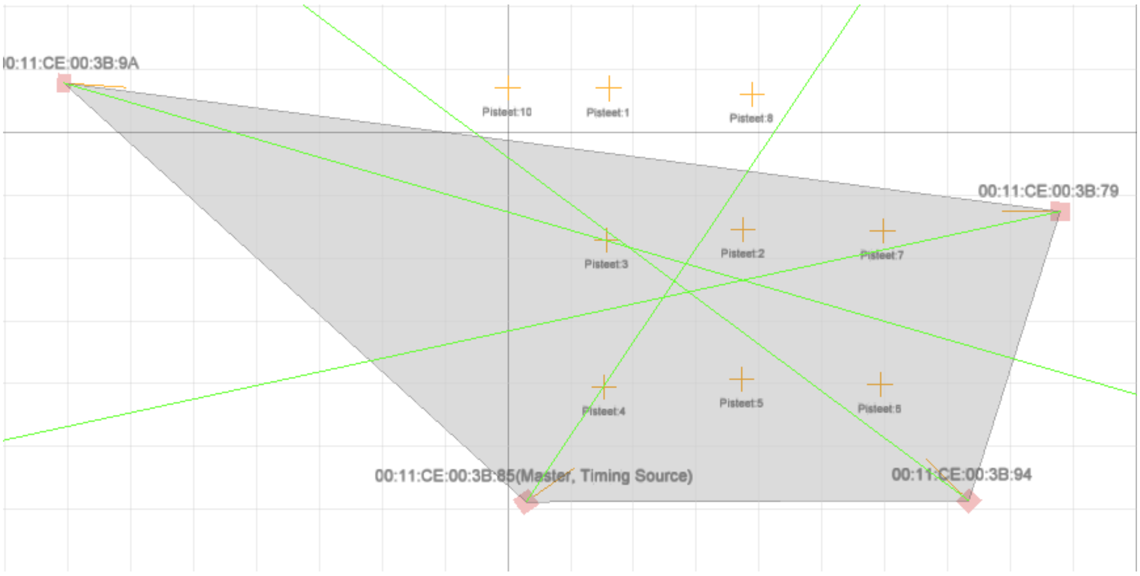


Figure 7.2: Ubisense sensor network from Ubisense location engine configurator software, showing the relative locations of the sensors

7.3.2. MultiTouch UI

The MES user interface was done via single 42 inch MultiTouch display by MultiTouch Ltd. The display works also as a touch screen, so the whole user interface (input and output) of the MES resides in the single screen. The novelty of the MultiTouch touch screen is as its name promises, that it enables practically unlimited simultaneous touch gestures to the screen. This makes it possible to use with both hands or even with multiple people, while making it much more convenient to use than only single touch enabling touch screens.

The touch screen functionality to the display is made with an infrared camera that resides inside the display box, and takes pictures of the screen. The feed from the camera is then processed, and bright “hand-like” shapes are deducted as hands and then transformed into controls and actions for the software.

The management execution system (MES) software is self made using the Cornerstone API supplied with MultiTouch. The MultiTouch display works as a normal computer screen, but in order to also use the input, the user interface must use the Cornerstone libraries (reusable pieces of code). These libraries mostly supply simple 2D visuals through basic shapes and image loading, and the focus of the Cornerstone library is on easy catching of user input and hand gestures.

The goal of the MES software is to enable simple monitoring and visualization of working orders, as well as adding orders and navigating product structures and processes. More about the software built on following chapter 7.4.

One of the goals of the project is to explore the possibilities of a modern touch screen UI in manufacturing environment, as currently the main area of touch screen usage is mobile device UIs and exhibition UIs for larger displays. The concept of multi-touch enables multiple users to collaborate around the single screen. This is a novelty regarding that computer interfaces have seldom had the possibility to use input devices simultaneously with multiple users in a same space (mouse, keyboard or single touch screens).

As such a multi-touch enabling touch screen is not by any means an integral part of a ubiquitous manufacturing system. In this project it was used to demonstrate the aforementioned possibilities of manufacturing interface, and to display the MES in a modern way.

7.3.3. RFID tags and readers

The RFID tags used in the project are regular passive tags. Two different set of tags were used, other set with the dimensions of 38mm x 13mm x 3mm, and the smaller tags with size of about 8mm x 8mm x 3mm. The functionality of the tags requires that the antennae inside the tags can not be in direct contact with metal, but both of the used tags had their internal antennae isolated, so that they could be installed on metal surfaces also. The tags are read using the standard European UHF (Ultra High Frequency) RFID frequency of 869 MHz.

The reader and that tags that were used in the project have a range of about 5 meters, but they are situated in the assembly station, where the RFID reader is only about 0.5 meters from the tags at most when the reading is required. See illustration of an assembly station at fig. 7.3. The assembly station in fig. 7.3 contains two wagons, one for component delivery and another for tool delivery. RFID antennae are situated in the station above the wagon slots, so that they can read the tags on tools and components.

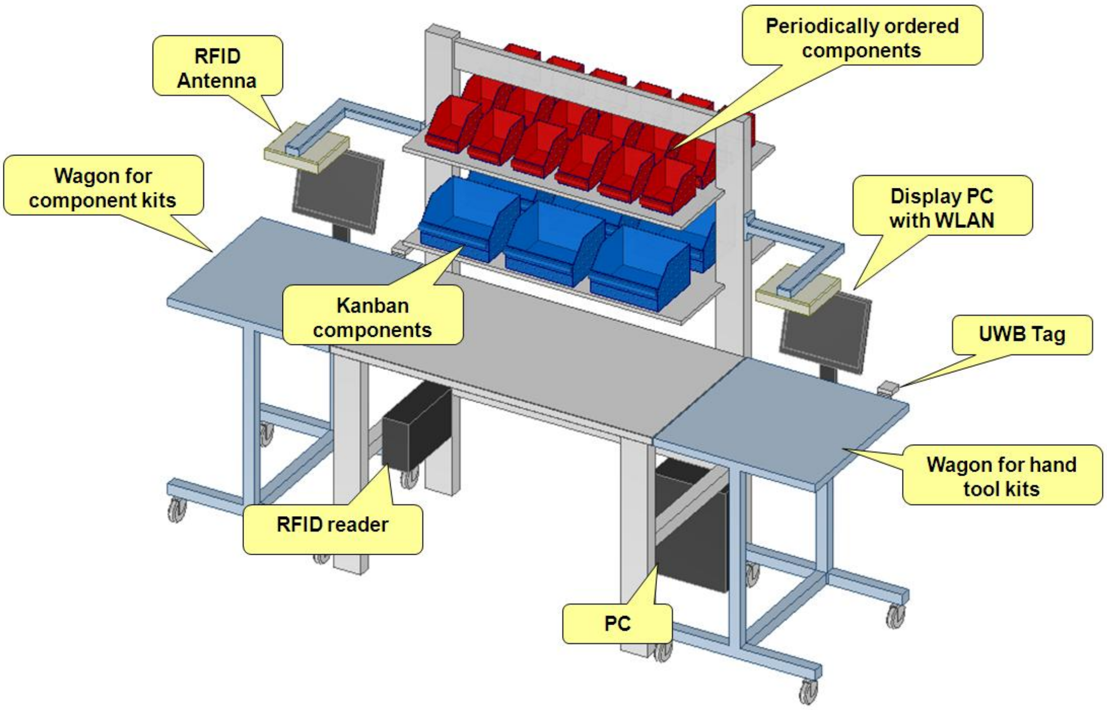


Figure 7.3: Illustration of an assembly station with RFID readers, a concept from Haag (2010, p. 5)

The RFID tag reading range varies greatly from the angle between the tag and the antenna, as well as from all reflecting and impassable surfaces and objects, such as metal tools in this case. This is the main reason why the antennae are situated very close to the wagons. The antennae are also directed towards floor and the kitting wagons, so that they get the minimal amount of readings from tags that they are not supposed to read, such as ones travelling by the assembly station.

7.4. Software suite

Most of the software used in the project was custom made, relying on the libraries and APIs available for aforementioned technologies.

As the system consists of multiple computers running various tasks, as the architecture revolves around central computer, which communicates with other nodes through TCP/IP socket messages. The high level architecture of the system is shown in fig. 7.4.

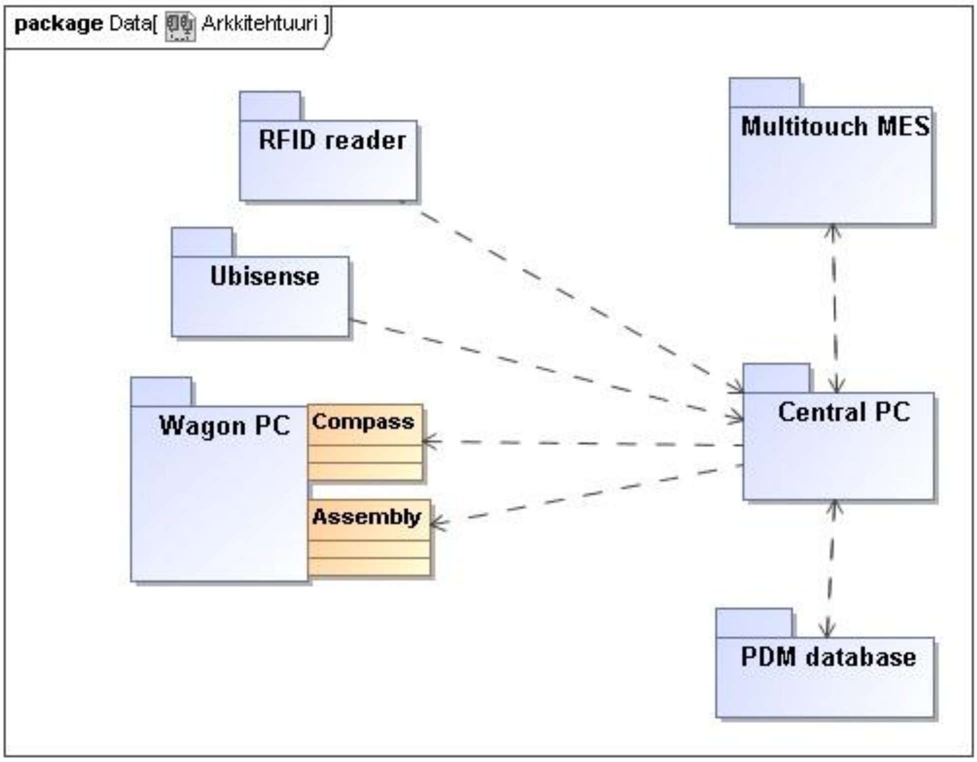


Figure 7.4: UManu system component architecture in UML

The functionality of the system is roughly that all the sensors (RFID and Ubisense) report to central computer where the monitored resources are. Then the central computer displays the monitored data in MES screen. MES also works as user input, where it is possible to manage work orders, which are then communicated to shop floor wagons that handle the kitting. The product and process information is contained in the PDM database that is accessed by the main computer to fetch required data.

The wagon PC has two main functions, which are to display a compass to help navigating in the warehouse to find the right components and tools, and an assembly helper that shows instructions for each step in manufacturing. The orders and the logic for wagon PC are situated in the main computer, that only supplies wagon PC with output.

Due to the closed nature of the test environment, and easier implementation, the server centric architecture for ubiquitous computing was chosen. The chosen technologies also better serve this approach over the approach of autonomous objects, as the various systems used are not interoperating by default, but instead need central software to merge their functionality. This makes it easier to use central computer as a middleman that collects all the messages from various systems (such as RFID reader and Ubisense) and then forms them all into sensible information and visualizes them. Also the research and demonstration nature of the software makes it more sensible to tie the functions into a central computer for better visibility and controllability of the system.

8. CASE RESULTS

This chapter discusses the completion of the case project. The utility and functionality of the case production line are described and evaluated as results. Also the suitability of the ubiquitous manufacturing product model is discussed.

8.1. Completion of the project

One of the main research topics was monitoring of the tools and kitting in the manufacturing environment. The technologies described earlier in chapter 7.3 were used so that RFID tags were implanted into resources, such as tools and machinery, and Ubisense UFB tags were used to track humans and kitting wagons. The concepts of interest were to see if it is possible to help the shop floor worker by displaying manufacturing instructions and the concept of warehouse navigation, where the kitting wagon gets navigation instructions via located components.

Another concept of interest would be to assess the usefulness of the transparency and data collection from the production line. But the sensibility and rewards from these actions vary greatly from case to case, so it is difficult to evaluate them through a single manufactured demonstration case, as there is no actual manufacturing done on the production lab to compare with.

The UManu -project started by studying PDM trends and ubiquitous computing visions, and building concepts combining these two areas. The next thing was acquiring the technologies, and the integration testing of the technologies. After it was clear that all the used technologies would work together more or less seamlessly, it was time to start the database formation, and the software suite functionality designing and programming.

The database was first modelled as C++ objects to test it, and use it as pilot database while developing the MES visualization and control software. The Cornerstone library for MultiTouch display was rather graphics oriented, and did not contain any functionality for conventional windowing and forms such as usual Windows-oriented software. This required quite a bit of work to turn the graphic elements into sensible and manageable windows with textual content, such as the product structures. The database model generation also needed a couple of iterations to evaluate the actual functionality of the model. This quick testing through hard coded software classes helped to detect biggest faults in data models.

8.1.1. Evaluation of the ubiquitous manufacturing product model

The data model built earlier in chapter 6 was used in the case project as planned, but here are some notices about it made during the project.

First of all the modelling about the activities requiring resources seems too strict. An activity does not necessarily need a certain tool, but instead a service provided by the tool. For example, if there was a need to attach pieces together with a Phillips drive screw (+ -shaped head), a flat-bladed screwdriver could be also used to perform the task. This would mean that both the screwdrivers would offer the same service of fastening a Phillips drive screw, but the Phillips screwdriver would not offer the service of fastening a slot (or flat) screw. For another example an adjustable wrench could offer multiple services regarding the size requirement for a nut, whereas a regular wrench could only offer a single size service.

In this sense the SOA architecture could possibly solve this modelling of resource requirements, as the resources could offer services, which the activities would then require. So that in the example case the screw would require a service of fastening a slot head screw.

The model itself does not take into account configurable products, as it just models a single structure rigidly. To add this support there could be an abstraction that the generic, abstract product can have a component type as an element instead of an actual component. For example for a configurable colour, all the possible paint options (colours) would be from the same component type. For more complicated configurable modules, the switchable modules would just have to share a type, so that when the product is put into production and into singular level, the usable module would be chosen. Männistö et al. (1998, p. 1111) also remind that there lies a need to model and verify valid combinations, as the configurable parts might impose some requirements for other parts of the product as well. Example of this would be a product with mutually exclusive configurations, such as a TV with bigger speakers -option that cannot be housed inside the regular sized casing. To fulfil these needs in configurability, there would also be a need for the valid combination listing depending configurable parts, whereas the configuration legality could be checked.

The three level depth of activities (process-task-operation) was also found to be quite cumbersome. This was because the demonstration product structures were purposefully kept simple, and did not include enough meaningful phases to build a three level hierarchy. But if the activity hierarchy would be made flexible, the model could be further enhanced. In this case the product could relate to resources and parts through fewer activity levels, or if needed through even more activity levels (in case of more complex products).

8.2. Results and functionality

As the project aimed mostly to the conceptualizing and testing the idea of having ubiquitous functionality in a product line, the results and the functionality did not go through further evaluation of actual production performance. In this chapter however are the findings on the functionality of the pilot ubiquitous manufacturing system.

8.2.1. Wagon compass and warehouse navigation

The main functionality of the kitting wagon is to find the required components, load them on, and take them to the assembly station where they are required. At the assembly station the components are unloaded from the wagon.

In order to help the navigating of the wagon, there is a portable computer on the wagon, which displays the required parts, and also a map showing the target locations, or a compass that directs to targets (akin to car navigator). The compass is enabled by getting the information about the orientation of the wagon as well as the location. Because of the accuracy of the UWB positioning, the orientation of the wagon is generated by planting a single UWB tag to the both ends of the wagon. This way the location difference of these two tags tells which way the wagon is pointing. This compass functionality could of course be implemented with just one tag and car navigator style orientation detection, which presumes that the wagon always moves “head first”, and detects the orientation when the wagon starts moving. But navigating inside a warehouse and making constant stops to pick up items could be a rather challenging environment for making such assumptions. This makes the double tag - orientation tracking described earlier more robust and therefore more viable solution.

The wagon also could have its own RFID reader to verify that the right components are loaded on it, but implementation done with this project did not have this functionality. Instead it was assumed that the worker can find the component, and place it upon the wagon. This is assisted so that when the wagon enters target predefined area of the warehouse, the wagon computer shows a message to pick up the certain part, and then shows the next direction to go. When the wagon leaves the area of where the wanted component was, it is assumed that the component is onboard and the wagon is directed to next warehouse location, or in the case of last component to kit, to the assembly station.

The actual utility of the warehouse navigation in demonstration environment was rather small, as the size of the plant was small, and the amount of various components to find was also limited. But the functionality of navigation was as expected, so that it could help the worker to navigate to desired locations. In case of actual factory, with larger areas, more components and assembly stations, the utility of navigation can be seen especially in getting to know the plant, in cases such as training a new worker or in the event of plant reorganization. Also if the products contain lots of variance and configurability, where worker time to time needs to find more rarely used parts, the usability of navigation seems viable.

As the navigation is done on basis of the required components and tools, it also reduces the possibility that worker forgets to load certain objects to wagon, and then is forced to make another kitting round to pick up missing objects. This yet again eases the load on workers memory.

8.2.2. Assembly helper

Assembly helper is the functionality implemented to the assembling station. When the kitting wagon is brought to assembling station, the station PC shows instructions regarding the assembly process. These instructions include which tools and components to use in current phases, and media such as diagrams, pictures and animations or videos of execution the process. Visualization of the assembly station was earlier presented in fig. 7.3.

As the system knows the exact order of tool and component usage, it can deduct which phase is currently underway by the absence of components and tools from the RFID readers that are located on top of the wagons. Besides aiding the worker, the same data that defines the exact phase of manufacturing is used to visualize the state of the assembly process in MES.

The utility of assembly helper is similar to warehouse navigation, as the utility in demonstration environment was rather bleak, but seems of increasing usability when the complexity and configurability of products increases.

8.2.3. Work planning and monitoring through MES

Because of the demonstrative nature of the system, the MES is mostly used to visualize the ubiquitous functionality of the system. Otherwise the state and progress of assembly would be only contained and used in the process logic, and following them in monitoring purpose would be cumbersome.

The MES screen simply allows views into production queues of the assembly stations, and adding orders to them. MES functionality also allows to browse generic product structures and to choose products or subassemblies from there for manufacturing, making in-plant orders. Then it is possible to allocate resources such as assembly stations, wagons and workers to these orders. In the pilot product line this allocation is done manually, but in actual system it could be formed automatically. The orders formed here are sent to queue, and kitting wagons handle collection of components and tools according to these orders.

Besides the visual monitoring through MES, the work planning system holds some novelty as well. As the product is modelled with the depth of ubiquitous manufacturing model (as described in chapter 6), a single order of manufacturing an item can be dynamically allocated into production, and into worker instructions, based on availability of the resources.

8.2.4. High-resolution management and analytical applications

As the tested concepts are mostly operational, the case study can be more likely seen as a maturity test for the technologies and their co-operation in a small scale industrial environment. The analytical effects of high resolution management and business process decomposition can be much more far fetching than can be tried out in demonstration environment such as the one used in case, and therefore can not be evaluated.

More suiting test environment for analytical functionality of ubiquitous manufacturing concepts would be actual real plant. In real environment the analytical data would have something to compare to, and could be verified. More discussion about these analytical functionalities of ubiquitous manufacturing can be found in the following chapter 9.

9. CONCLUSION

This chapter summarises and discusses the study, its implications and assesses whether or not it was successful. Also the further research questions raised by the study are briefly discussed in the chapter 9.4, and a curt notion on how a company can start developing its IT towards ubiquitous system in the chapter 9.5.

9.1. Summary of the study

The literary review concludes that the field of product data management is very fragmented software wise. There are no widely accepted standards to rely upon, but instead various actors have their own or supply chain specific software. The same trend of poor standardization is also present in product data modelling, as there are no common standards either. The most common practice present seems to be using extensive mark-up language (XML) for implementing the models. This should guarantee at least low level semantics for the models and data transformations, but it does not implement interoperability between systems using different models.

Ubiquitous computing as a paradigm still holds lots of expectations and un-opened doors and possibilities, as there are not too many applications present yet, and even fewer in active use. Surely as the ideology of ubiquitous computing gains more footing, more and more applications will surface, some of which are applicable to manufacturing environment as well.

One of the main problems present also with ubiquitous computing currently seems to be the lack of standardization. This is crucial as the key ideology and the strength of ubiquitous computing lies in linking of objects and their communication. So having a single device with ubiquitous functionality offers a rather bleak experience of the ubiquity and possibilities of the technology. This further straightforwardly means that for the ubiquitous technologies to fully lock in, there is a need for multiple ubiquitous applications during the launch for them to co-operate and be viable at all.

For a single company such standardization can be seen as a rather monumental task, because ubiquitous technologies should be embedded into everyday things that vary greatly from each other, such as mobile phone, fridge or computer. In manufacturing context, this lack of standardization does not represent such a big risk or problem, as the manufacturing environment is much more closed and controlled than the “outside world” of customer goods. An example of this could be a single factory with ubiquitous functionality, which would not really suffer if the factory next door does not use the same ubiquitous manufacturing standard. This means that in terms of standardization, implementing ubiquitous computing concepts into manufacturing should be much easier than into customer goods.

The ubiquitous manufacturing concepts discussed in this study were closely related to digital and automated factory concepts, with the focus of increasing the amount of knowledge in products and processes. The key functionality is the digital representation of products, where the whole product process is automatically rendered digitally, in addition to occurring in real life.

Digitalizing and adding knowledge and functionality just into products still leaves the manufacturing line rather unbalanced in terms of actual functionality. The digital function should also include processes and other objects and resources of the manufacturing plant, in order to have something that the products can interact with. This integration can be seen as the most important point of ubiquitous manufacturing, binding together the products and processes. Nevertheless this integration only forms the foundation to build ubiquitous functionality and logic upon.

Meyer et al. (2006, pp. 6-8) defined the lack of technology, standardization and data integration as the main obstacles for computer-integrated manufacturing and digital factories. Thus filling these three gaps can be seen as a high level requirement for any ubiquitous manufacturing system. On basis of the study it is sufficient to say that clearly the technologies to enable this ubiquitous manufacturing already do exist, and the complexity of these systems can be eased to some extend by architectural choices such as SOA. The problems with data integration can possibly be tackled with ontologies. As ontologies are already used to integrate scientific data of varying topics, one would think that manufacturing and its information needs are not too heterogeneous fields to integrate either. Out of these three obstacles, only standardization still remains problematic. In closed manufacturing systems the standardization problem can be partially skipped. But as there probably are needs present to manufacturing information systems to co-operate with systems of other actors in the supply chain, such as supplier’s systems, the problem yet persists.

Putnik et al. (2007, p. 3205) state risks for ubiquitous manufacturing implementation being knowledge and trust management, complexity, legal issues, social awareness and acceptance. The usage of ubiquitous technologies can be clearly seen as a risk for personal privacy, as the location and actions of humans that work in the ubiquitous systems can be tracked as well. Besides knowledge management, these issues mentioned by Putnik et al. (ibid.) however are not PDM oriented, and thus not discussed in this study.

9.2. Discussion and implications

The study about concepts enabled by ubiquitous manufacturing rather clearly seems to conclude that the ubiquitous manufacturing paradigm can be viable and profitable for companies that can exploit it successfully. Naturally it cannot be whole-heartedly recommended for every company, but then again very few information systems can be. Thus the sensibility of ubiquitous manufacturing should be further explored in each individual field of manufacturing in order to form area-specific recommendations.

At least the possibility of ubiquitous technology to aid manufacturing should be kept in mind when designing new product data modeling, or production plant actions.

9.2.1. High resolution management through process decomposition

The system with all central objects and processes tracked generates vast amounts of information, which can be further used to enable “high resolution management” as discussed earlier in chapter 4.4. As the result of the study regarding information and knowledge management, this can be seen as one of, if not the biggest single effect of ubiquitous manufacturing.

The high resolution real-time information is also present in the processes, which enables using computer logic for allocating the resources of the product line and optimization for the current priorities and requirements of production. An example of this would be an urgent order that could be shot through the production without queuing to various stages of the production. This high resolution information can also be generated automatically from the processes, thus there is no need to have external measurements on the functionality of the system. The same information is used for both operative and analytical functionality.

In analytical utility the main advantages this abundance of information leads into are enhanced decision making and more controllable manufacturing. Besides allowing more detailed view to manufacturing, the information generated by ubiquitous manufacturing can potentially allow finding previously hidden details.

9.2.2. Application to the environment

Gill & Cormican (2006, p. 2) define that SME companies in general are dynamic, flexible and innovative, but not always at the cutting edge of the technology. This leads to believe that the SME companies have the capability to adapt into ubiquitous manufacturing environment, but do not necessarily work as large scale pioneers. Another obstacle regarding this is the SME companies’ lack of force to drive standardization on these concepts. More force for inter enterprise standardization could be gained through coalitions of SME who would collectively begin using similar ubiquitous systems. More likely scenario for these technologies to become more popular would be single big manufacturers starting to use some standards, and forcing their suppliers to comply with them as well. Haller et al. (2008, p. 14) also believe that companies are currently reluctant to adopt these new technologies, and waiting to see profitable real world deployments first.

The declining prices of the technologies required for the location services and wireless technologies make it possible to include at least some location aware functionality into product line without big investments in the hardware. The software side implementation for the system could probably be more difficult task, if the systems need to be tailored for every occasion and there are no off-the-shelf or open source solutions available. This is likely going to be the case for now, as the ubiquitous functionality is still a novelty, and not well established standard in manufacturing systems.

Also in the case study it became evident that the ubiquitous manufacturing concepts are well suited for complex product structures, where various different operations need to be done. This is clearly in consensus to respond the challenges of product data management, with the current notion that products are constantly increasing in complexity.

9.2.3. Addition to the knowledge

The key addition to the knowledge of the area from this study seems to be the concept of adding the layer of managing concrete, real world objects into PDM systems. While the business process decomposition aims for mapping of the processes and activities as well (as discussed in Lanz 2010 and Haller et al. 2008), this would seem as a possible place to add the singular level object management as well. The idea of singular level item management has probably been around as long as PDM systems have been given any thought, but by now the technology should be able to facilitate these concepts. Even if the modern computing still would lack the strength to pull off the industrial level performance, the concepts can be tested out in smaller scale and made ready for the technology to catch up.

The usage of semantic ontologies to the field of product data management and product lifecycle management seems also like a viable application. A single addition to knowledge could be the evaluation of product related information or the usage of ontologies to model product related data done in chapter 5. A notion should also be made about some similar analogies in production and computer science. These analogies being that some computer science concepts and paradigms at least seem to fit also in manufacturing environment. As such SOA and semantic ontologies were discussed in this study.

9.3. Assessment of the study

The primary objective of the study was to explore the relation between ubiquitous technologies and the product data management in manufacturing context. This objective was clearly met, but definitely the relation is not extensively described. The ubiquitous manufacturing concept tested in this study is just a single possibility offered by the ubiquitous computing paradigm. Clearly ubiquitous computing holds lots of possibilities in enterprise context outside manufacturing phase also. It could greatly enhance also the later phases of product lifecycle management (PLM), but also in other fields than just manufacturing industry.

The empirical part could have been more accurate on putting the ubiquitous manufacturing model to test, but as the model and concepts were developed simultaneously, they largely support each other at default. Situating the testing environment in actual case companies could have been fruitful, and interviews could also have better shed light on the current state of PDM systems and their requirements.

Nevertheless building an actual ubiquitous, although simple, system helped a lot in understanding the main principles of the ideology. This testing also helped to generate concrete requirements for the data model in addition to ones found in literature research earlier. Also the requirements for ubiquitous manufacturing were found and met at least in the case study, as the pilot product line clearly had some ubiquitous functionality, and no bigger issues were found in the evaluation of the ubiquitous product model.

Also it seems that the guidelines set for the study in chapter 1.3 were met, as in retrospective none of them seems totally unapplied to the study. Although the contribution to the research area can be seen as rather slim, I believe that the contribution that is expected from a master’s thesis is met. The study also formed according to IS research model by Hevner et al. (2004, pp. 78-81), as the IS model was first built by fusing knowledge base and environment, and then evaluated with a case study.

9.4. Possibilities and further studies

As seen in this study, the ubiquitous manufacturing paradigm allows very wide variety of possibilities to enhance the production. This chapter gathers up some of the ideas enabled by ubiquitous manufacturing, that came up with the empirical section of the study.

9.4.1. Hybrid positioning

The positioning technologies can be bundled up in some extend to work around their shortcomings. For example putting a large UWB tag to all items is highly impractical due to its cost and size, and RFID tags do not offer such seamless positioning with basic RFID readers.

Nevertheless in some cases, such as the kitting wagon, it is clearly possible to combine these two, so that the wagon itself is contains UWB tag and also a low power RFID reader. Then the wagon knows exactly what objects are onboard by reading their RFID tags, and we can accurately tell the location of all the RFID tagged components or tools that travel with the wagon, by the positioning of the wagons UWB tag.

9.4.2. Human in ubiquitous manufacturing

The ontologies such as one presented by Lanz (2010) and the ubiquitous manufacturing product model here aim to define the product in sum of actions. The focus constantly here being the modelling of product, but the process decomposition also allows very accurate modelling of the human actions in the product line. This can be used to generate work orders for each individual worker that take into account the straining of the worker.

The real time constant measuring enabled by traced objects also allows more accurate measuring of the workers performance and the possibility to detect hazardous situations through key figures in performance. Also the dynamic instructions and generally intelligent objects in the product line can potentially shift the role of a shop floor worker, by cropping such obsolete tasks like remembering and choosing objects.

Gill & Cormican (2006, pp. 5-6) point out that ambient intelligence systems should be very human centric. This human focus can even go to extend the system to take into consideration its user’s habits, needs, gestures and emotions (ibid.). As discussed in chapter 2, another area of ubiquitous computing is human centric interfaces and interaction, but this study has not really taken these into account. Zuehlke (2010, pp. 129-130) also noted that some of the earlier attempts to standardize computer integrated manufacturing (CIM) have failed mainly because of the lack of consideration towards human factors.

9.4.3. Extending the ubiquitous manufacturing model to cover PLM

Another big area that was cropped out of the study was all the later phases of products lifecycle after manufacturing. With such a detailed look to product by the he end of beginning of life (BOL), it could be possible to build some PLM applications on top of the defined system and BOL data. But surely a comprehensive approach, answering all the further requirements posed by later phases of lifecycle, would enable much more functionality for the product data management.

Especially with products with long lifecycles, such as heavy machinery, the benefits of extended data model should become evident. Example cases that could benefit from exact product data being available could be in maintenance and service areas.

9.4.4. Generic modelling for UM operational systems

While the ubiquitous manufacturing model defined in this study is keen on modelling what are the objects on the product line, it does not facilitate the actual functionality of the system in production. Meaning that no matter the depth of data, there will not be any use for it without an operational system handling the product and process data.

An interesting area for research could be to find out if there is a possibility to make a generic architecture and process modelling for the ubiquitous manufacturing functionality as well. Clearly it is the functionality of the operational systems in ubiquitous manufacturing that enables most of the new concepts, while the data model just works as a foundation. This functionality modelling along with UM data model would work as a simple model for the whole PDM system in manufacturing.

9.5. First steps towards ubiquitous manufacturing

The ubiquitous manufacturing concept seems to be very scalable, which would transforms into easier implementation and step by step adaption of components. As discussed earlier, it might not be possible for a company to digitalize all the processes, such as machinery that does not have CNC interfaces (Computer Numerical Controller) or PLC interfaces (Programmable Logic Controller). But it might be possible to digitalize all the product / component functions anyway, and use that information to generate ubiquitous functionality.

The ambient intelligence generated through positioning objects can be used to some extend to imitate interaction with digitalized processes. An example of this would be that in the conveyor belt the location of the object will likely be able to tell something about what phases and processes has the object already gone through.

The requirements list generated in chapter 5 can be seen as a viable starting point or as a crude checklist for things to consider when defining PDM functions that can offer some basic level support for future ubiquitous manufacturing functionality. Apart from the information systems development, Putnik et al. (2007, p. 3205) lists development of specific organizational architectures, protocols, meta-organisational environments and legal issues as recommended measures for ubiquitous manufacturing development. The development of these non-technical factors will most likely be in place, as the shift to ubiquitous manufacturing is likely to have permanent effects on ways of working and manufacturing.

BIBLIOGRAPHY

Abawajy, J.H. 2009. Human-Computer interaction in ubiquitous computing environments. International Journal of Pervasive computing and Communications, vol. 5, no. 1. pp. 61-77.

Brunner J., Ma L., Wang C., Zhang L., Wolfson D., PanY, Srinivas K. 2007. Explorations in the Use of Semantic Web Technologies for Product Information Management. WWW 2007, pp.747-756.

Diegel, O., Bright, G., Potgieter, J. 2004. Bluetooth ubiquitous networks: seamlessly integrating humans and machines. Assembly Automation, vol. 24, no. 2, pp. 168-176.

Garfinkel, S. A Less Personal Computer. Technology Review, vol. 113, no. 3. pp. 86-87.

Gill, S., Cormican, K. 2006. Support ambient intelligence solutions for small to medium size enterprises: Typologies and taxonomies for developers. 12th International Conference on Concurrent Enterprising, Milan, Italy. 8 p.

Gruber, T. 1993. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, vol. 5 nr. 2. pp. 199-220.

Haag, M. 2010. Ubiquitous Assembly Station Concept. 18 p. VTT internal document, limited availability.

Haller, S., Karnouskos, S., Schroth, C. 2008. The Internet of Things in an Enterprise Context. Future Internet – FIS 2008: First Future Internet Symposium, revised selected papers. Springer-Verlag. 15 p.

Hevner, A., March, S., Park, J., Ram, S. 2004. Design Science in Information Systems Research. MIS Quarterly, vol. 28, no. 1. pp. 75-105.

Johanssen, J., Madsen, O., Jensen, H. V., Vestergaard, A. 2010. Manufacturing 2025 – Five future scenarios for Danish manufacturing companies. Aalborg University. 36 p. Limited availability.

Kaiser, W., Pottie, G., Srivastava, M. 2003. Networked Infomechanical Systems (NIMS) for Ambient Intelligence. UCLA Center for Embedded Networked Sensing, Technical Report 31, December 2003. 20 p.

Kortelainen, J., Mikkola, A. 2010. Semantic data model in multibody system simulation. Proceedings of the Institution of Mechinical Engineers, Part K: Journal of Multi-body Dynamics, vol. 224, no. 4. Sage Publications. pp. 341-352.

Kovacz, Z., Le Goff, J.-M., McClatchey, R. 1998. Support for product data from design to production. Computer Integrated Manufacturing System, vol. 11, no. 4. pp. 285-290.

Kranz, M. Holleis, P., Schmidt, A. 2010. Embedded Interaction: Interacting with the Internet of Things. Internet Computing, IEEE, vol. 14, no. 2. pp. 46-53.

Kropsu-Vehkaperä, H., Haapasalo, H., Harkonen, J., Silvola, R. 2009. Product data management practices in high-tech companies. Industrial management & Data Systems, vol. 109, No. 6. pp. 758-774.

Lanz, M. 2010.Logical and Semantic Foundations of Knowledge Representation for Assembly and Manufacturing Processes. Tampere University of Technology. Tampere. 138 p.

Loshin, D. 2009. Master data management. Morgan Kaufmann publishing, USA. 274 p.

Lucke, D., Constantinescu, C., Westkämper, E. 2008. Smart Factory, - A Step towards the Next Generation of Manufacturing. The 41st CIRP Conference on Manufacturing systems. pp. 115-118.

Lucke, D., Constantinescu, C. 2010. Smart Factory Data Model: Foundation of Context-Aware Applications for Manufacturing. CIRP ICME ´10, 7th International Conference on Intelligent Computation in Manufacturing Engineering (23-25 June 2010). 4 p.

Luh, Y. P., Chu, C. H., Pan, C. C. 2010. Data management of green product development with generic product architecture. Computers in Industry vol. 61, pp. 223-234.

Ma, J., Yang, L., Apduhan, B., Huang, R., Barolli, L., Takizawa, M. 2005. Towards a Smart World and Ubiquitous Intelligence: A Walkthrough from Smart Things to Hyperspaces and UbicKids. Journal of Pervasive Computing & Communication, March 2005. Troubadour Publishing ltd. pp. 53-68,

McGovern, J., Sims, O., Jain, A., Little, M. 2006. Enterprise Service Oriented Architectures. Springer Netherlands. 408 p.

Meyer, H., Fuchs, F., Thiel, K. 2009. Manufacturing Execution Systems: Optimal Design, Planning and Deployment. McGraw-Hill. 249 p.

Männistö, T., Peltonen, H., Martio, A., Sulonen, R.1998. Modelling generic product structures in STEP. Computer-Aided Design, vol. 30, no. 14. pp. 1111-1118.

Ou-Jang, C., Chang, M. J. 2006. Developing an agent-based PDM/ERP collaboration system. International Journal of Advanced Manufacturing Technology, vol. 30, no. 3-4. pp. 369-384.

Peltonen, H., Martio, A., Sulonen, R. 2002. PDM - Tuotetiedon hallinta. Edita Prima Oy, Helsinki. 169 p.

Putnik, G., Cardeira, C., Leitao, P., Restivo, F., Santos, J., Sluga, A., Butala, P. 2007. Towards Ubiquitous Production Systems and Enterprises. IEEE ISIE 2007. pp. 3203-3208.

Putnik, G., 2010. Ubiquitous Manufacturing Systems vs. Ubiquitous Manufaturing Systems: Two paradigms. The 36th CIRP-International seminar on Manufacturing systems. 03-05 June 2003, Saarbruecken, Germany. 2 p.

Saaksvuori, A., Immonen, A. 2008. Product Lifecycle Management. Springer-Verlag Berlin and Heidelberg GmbH & Co. 260 p.

Sackett, P.J., Bryan, M. 1998. Framework for the development of a product data management strategy. International Journal of Operations & Production Management, vol. 18, no. 2. pp. 168-179.

Satyanarayanan, M. 2001. Pervasive Computing: Vision and Challenges. IEEE Personal Communications, vol. 8, no. 4. 10 p.

Shadbolt, N., Berners-Lee, T., Hall, W. 2006. The Semantic Web Revisited. Intelligent systems, IEEE, vol 21, No 3. pp. 96-101.

Subirana, B., Sarma, S., Fleisch, E. 2006. High-Resolution Management. IESE Alumni Magazine, Juli/September 2006, pp. 8-13.

Sudarsan, R., Fenves, S.J., Sriram, R.D., Wang, F. 2005. A product information modelling framework for product lifecycle management. Computer-Aided Design 37 (2005). pp. 1399-1411.

Suh, S.-H., Shin, s.-J., Yoon, J.-S., Um, J.-M. 2008 UbiDM: a new paradigm for product design and manufacturing via ubiquitous computing technology. International Journal of Computer Integrated Manufacturing, vol. 21, No 5. pp. 540-549

Thomas, V., Neckel, W., Wagner, S. 1999. Information technology and product lifecycle management. Proceedings of the 1999 IEEE International Symposium on Electronics and Environment. pp. 54-57.

W3C, World Wide Web Consortium. 2010. www.w3c.org (read 22.11.2010)

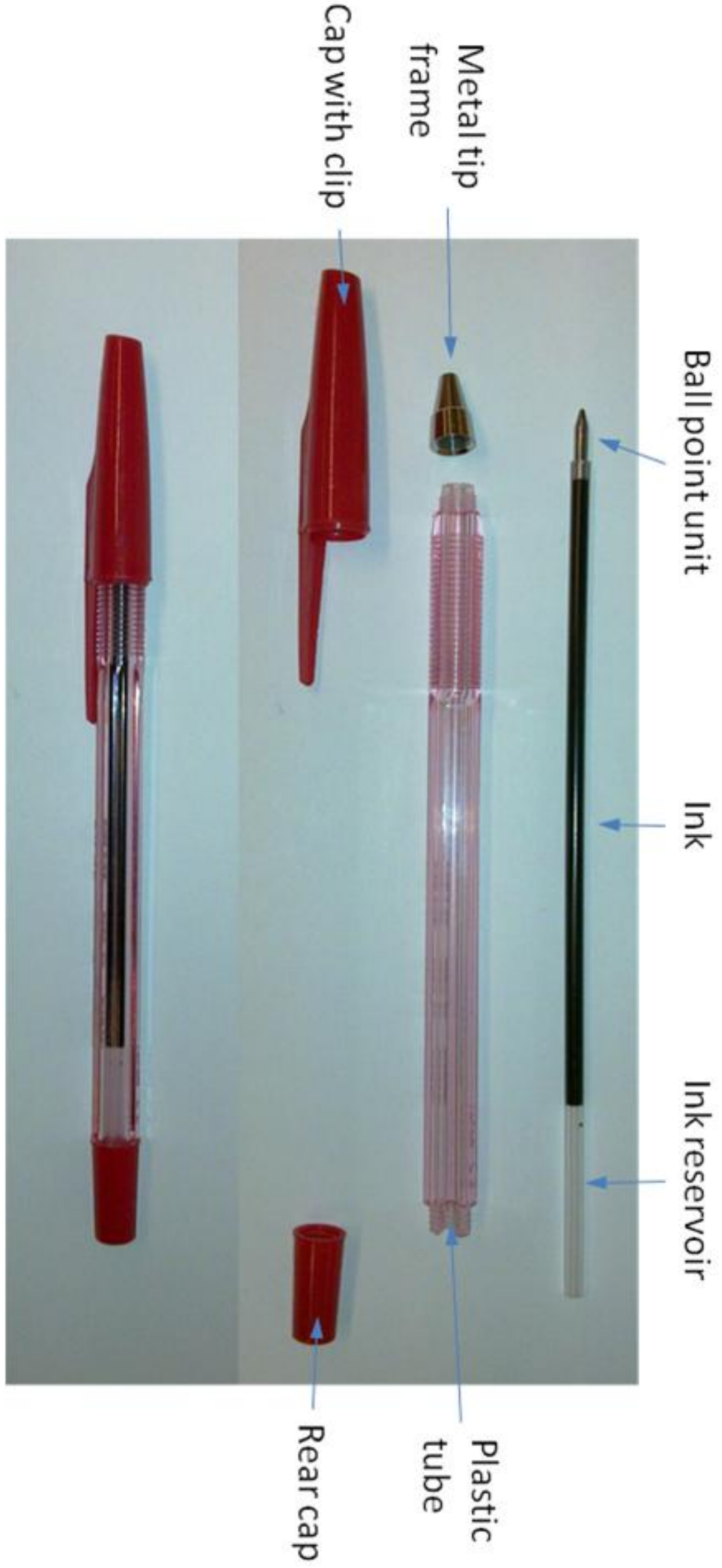
Weiser, M. 1991. The Computer for the Twenty-First Century. Released in Scientific American. September 1991. Available from [http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html] (read 21.5.2010)

Westkämper, E., Jendoubi, L. 2003. Smart Factories, manufacturing environments and systems of the future. The 36th CIRP-International seminar on Manufacturing systems. 03-05 June 2003, Saarbruecken, Germany. 5 p.

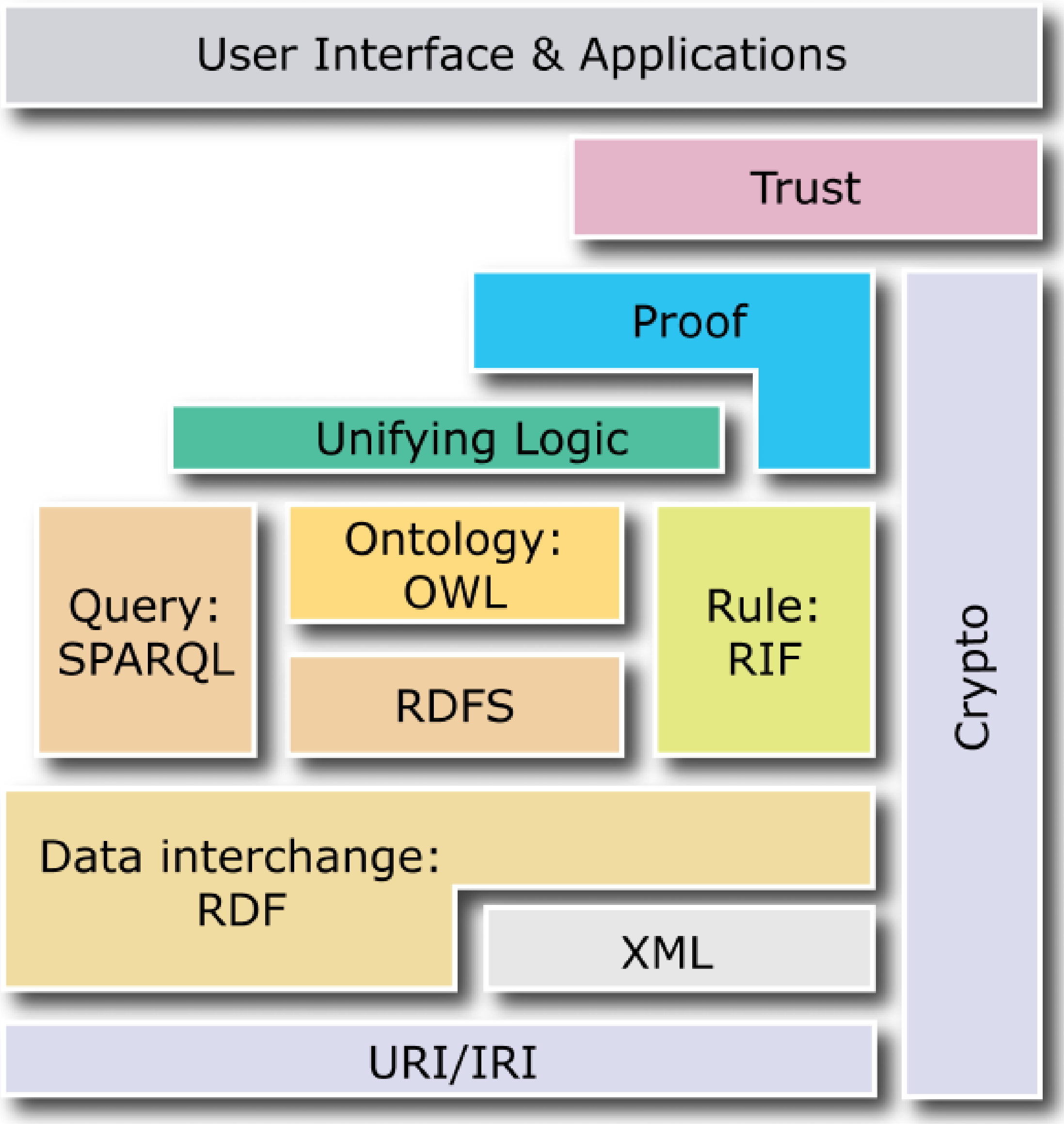
Youseff, L., Butrico, M., Da Silva, D., 2008. Toward a Unified Ontology of Cloud Computing. Grid Computing Environments Workshop, 2008. GCE’08. 10 p.

Zuehlke, D. 2010. SmartFactory – Towards a factory-of-things. Annual Review in Control. Vol. 34, no. 1. pp. 129-138.

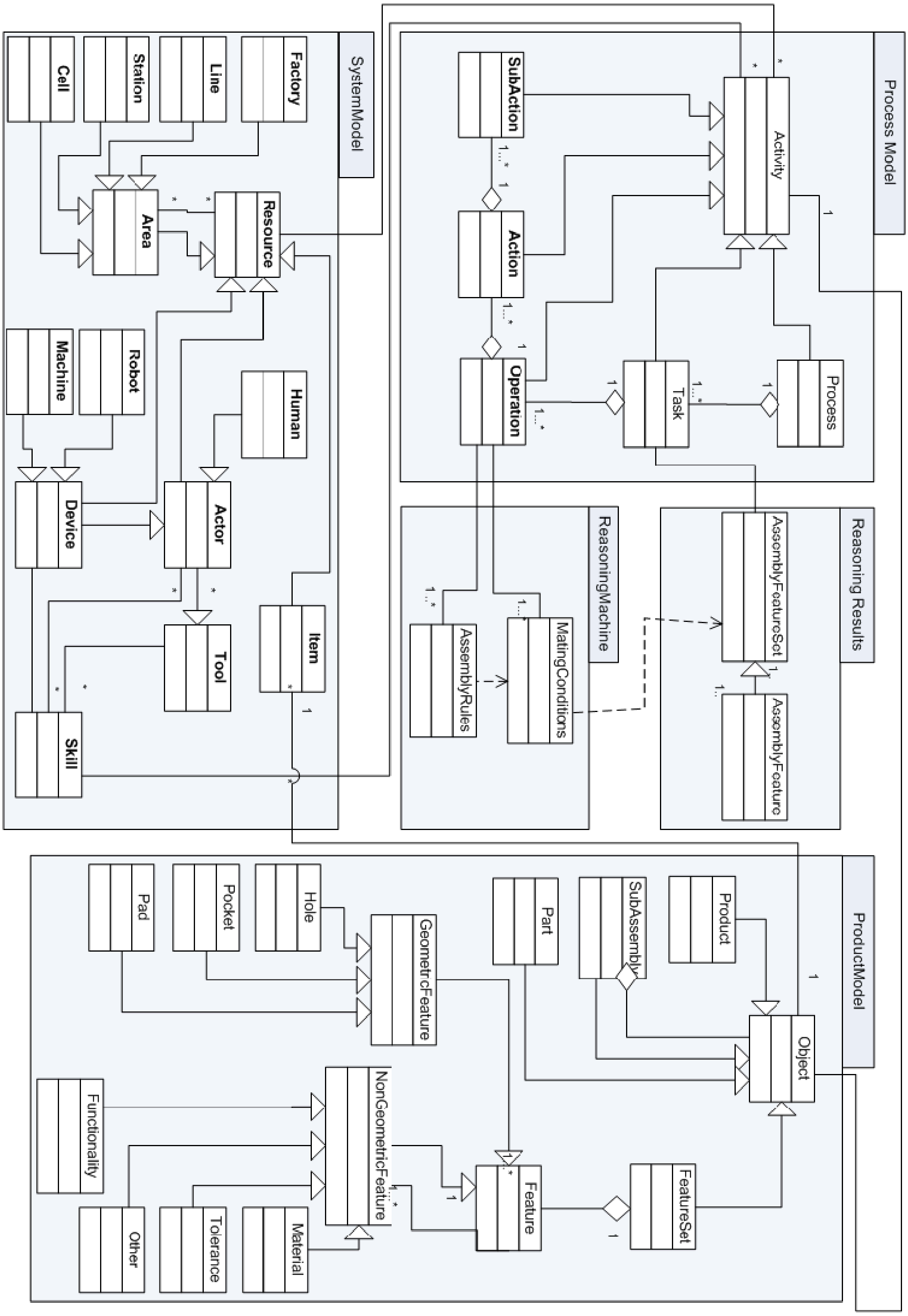
APPENDIX A: Ballpoint pen structure



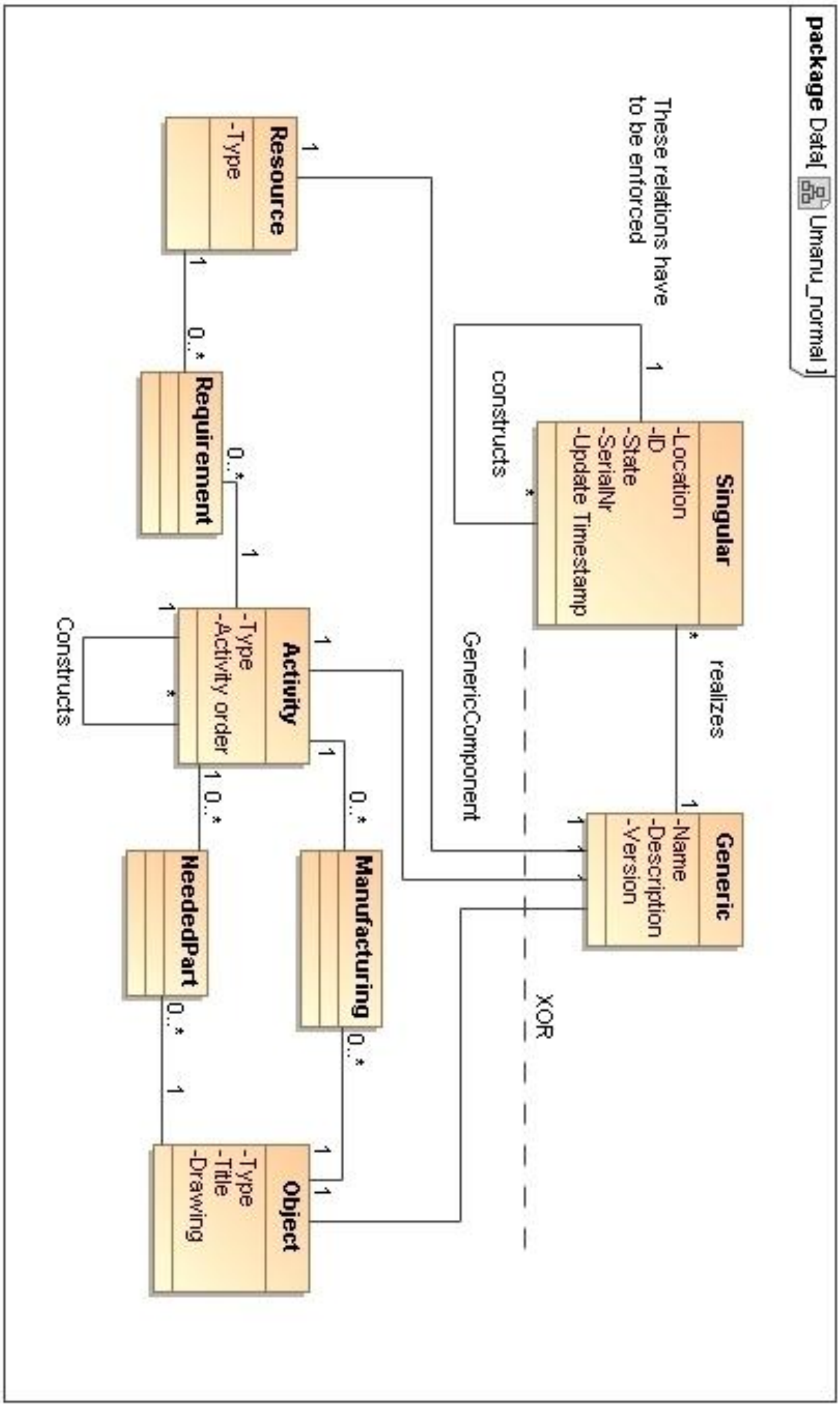
APPENDIX B: The semantic web technology stack from W3C (2010)



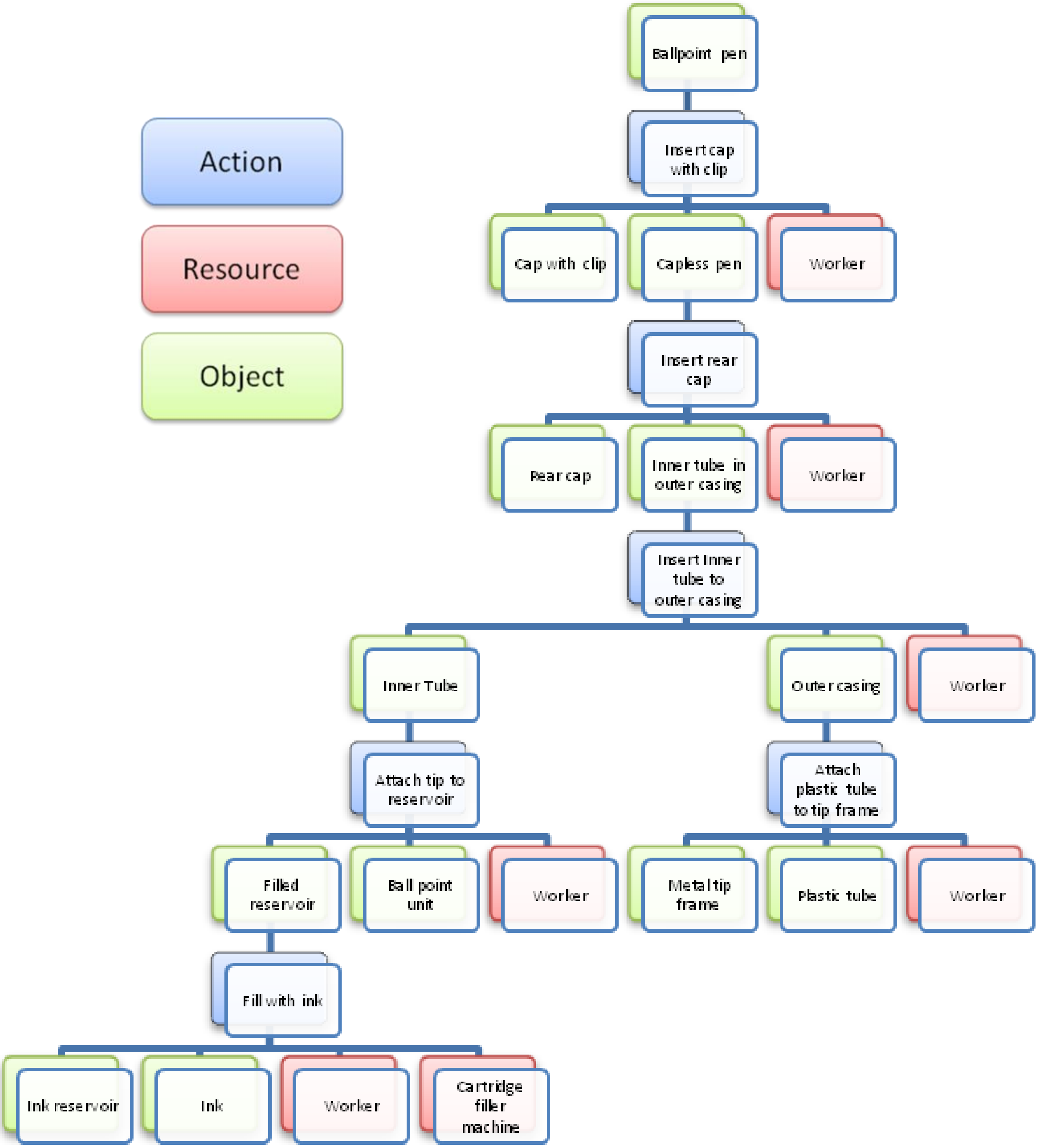
APPENDIX C: Product-Process-System –model by Lanz (2010, p. 65)



APPENDIX D: SQL compliant version of the Ubiquitous Manufacturing Model



APPENDIX E: Ballpoint pen modelled with ubiquitous manufacturing product model



APPENDIX F: Manufacturing Lab Blueprint