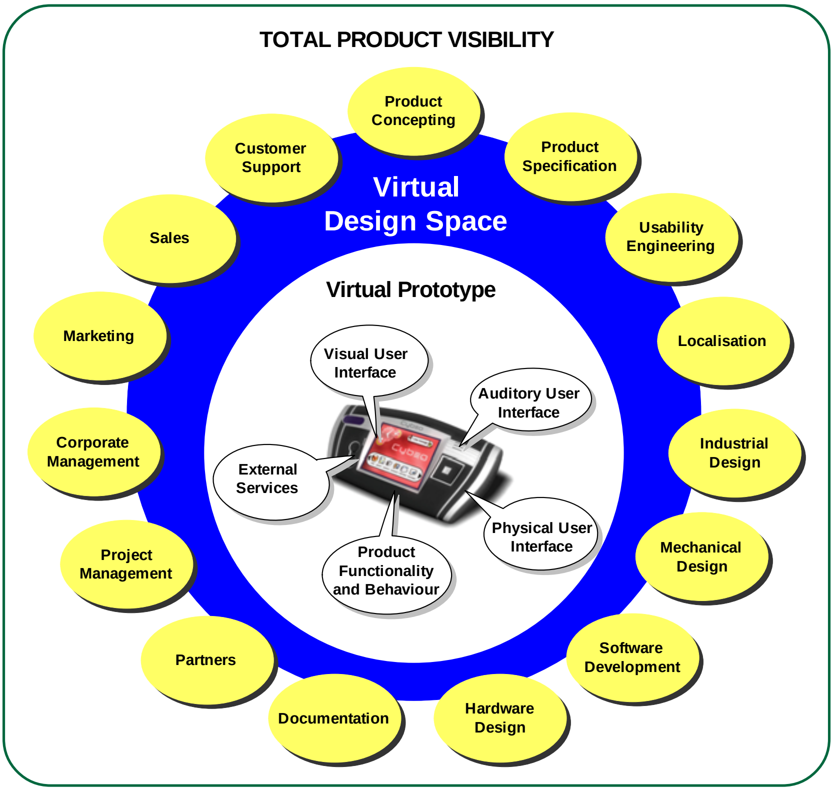
ESPOO 2006

VTT PUBLICATIONS 615



Mikko Kerttula

Virtual Design

A Framework for the Development of Personal Electronic Products

VTT PUBLICATIONS 615

Virtual Design

A Framework for the Development of Personal Electronic Products

Mikko Kerttula

Academic Dissertation to be presented,
with the assent of the Faculty of Technology, University of Oulu,
for public discussion in Auditorium L10, Raahensali, Linnanmaa,
on September 29th, 2006, at 12 o'clock noon

ISBN 951–38–6863–X (soft back ed.)
ISSN 1235–0621 (soft back ed.)

ISBN 951–38–6864–8 (URL: http://www.vtt.fi/publications/index.jsp)
ISSN 1455–0849 (URL: http://www.vtt.fi/publications/index.jsp)

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 3, PL 1000, 02044 VTT
puh. vaihde 020 722 111, faksi 020 722 4374

VTT, Bergsmansvägen 3, PB 1000, 02044 VTT
tel. växel 020 722 111, fax 020 722 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 3, P.O.Box 1000, FI-02044 VTT, Finland
phone internat. +358 20 722 111, fax + 358 20 722 4374

Technical editing Anni Kääriäinen

Valopaino Oy, Helsinki 2006

Kerttula, Mikko. Virtual Design. A Framework for the Development of Personal Electronic Products. Espoo 2006. VTT Publications 615. 218 p.

Keywords virtual design, virtual prototyping, virtual prototype, product development, personal electronics, information appliances, development process, concurrent engineering

Abstract

Personal electronic products are electronic devices that people use for their everyday tasks. They have customizable features and offer services that are mostly implemented with software and controlled with software-based user interfaces. The development of personal electronics is a demanding task for companies. It introduces a design target that requires multi-discipline co-operation and expertise. Besides for the integration of more traditional development areas of mechanical, hardware and software engineering, the demand for customer satisfaction extends the challenge with new expertise areas that focus on good usability and an appealing design of the product. In general, more effort must be placed on the early design phases that validate the correct product decisions that are necessary for a successful business in the international markets.

This thesis introduces virtual design as a product development framework for the personal electronics industry. Virtual design is based on the effective use of realistic and functional product simulations called virtual prototypes over the different product life cycle phases. The goal of this thesis is to study how virtual design can help companies in developing personal electronics more efficiently, i.e. with better customer satisfaction, reduced effort and shorter development time.

The research work is started by analysing the general challenges that usually are met during the development of a personal electronic product. Next, the development processes and tools that are usually applied in the development of electronic and telecommunication products are studied, along with their typical limitations especially regarding the development of personal electronic products. The analysis is founded on experience from industrial cases. Real case studies are also applied in defining the requirements for better product development and virtual design ultimately aiming to fulfil these requirements.

The potential of virtual design in the development of personal electronic products is evaluated by way of three different cases. The cases consist of 2 public cases and a summative case that describes the development of mobile phone user interface software with virtual design. The cases concretize the results of virtual design in the specific design tasks and also as a wider product development process framework.

This thesis demonstrates that virtual design supported by appropriate tools can improve the performance of a company developing personal electronic products. Maximum benefit can be attained if virtual prototypes are used throughout the entire product life cycle, from concept design to marketing and customer support. It is then that the concrete advantages of the approach, such as improved communication between the design team members, reduced development time with parallelized design tasks, reduced costs based on component reuse, and better product targeting based on improved concept validation, will become even more evident.

To my father, Erkki, who passed away in summer 2005,
during the most hectic writing period of this thesis.

Preface

The work conducted for this thesis was done in two main phases. The first period was at VTT Technical Research Centre of Finland, between 1996 and 2001, and then the work was taken up again to be revised and updated between 2004 and 2006 based on my industrial experience at Cybelius Software Ltd. and CCC Group.

At VTT, the research was mainly conducted in a national research project on Virtual Prototyping Services for the Electronics and Telecommunications Industries (VIRPI) during 1996–1999. The objective of the project was to diffuse virtual prototyping technology to industry and further develop it according to the feedback received from companies. The project was funded by Tekes, the Finnish Funding Agency for Technology and Innovation, and industrial and research partners. VIRPI project proceeded in close co-operation with VTT's strategic Virtual Reality Prototyping (VRP) project (1996–1998), which was responsible for VRP technology development at VTT. Together with the VRP project and its industrial co-projects, where VRP technology was adapted to industrial needs, VIRPI provided a fruitful base for the research of virtual prototyping and virtual design. It focused, especially, on the areas of concept and detailed product design (software and mechanics), web-marketing and usability engineering.

The work at VTT was further broadened in various national research projects such as CyPhone (Personal Virtual Services Based on Picocellular Networks, co-funded by Tekes), PAULA (Personal Access and User Interface for Multimodal Broadband Telecommunication, co-funded by the Academy of Finland) and VRFlow (Virtual Reality and Hypertext Functionality Enhanced Webflow for Consumer Behavior in Electronic Commerce Networks, funded by Tekes), which extended the research areas to user interface and value-added services (VAS) design, demonstration of mobile terminals, and to e-commerce and e-business. Additionally, I had an interesting opportunity for exchange of ideas within the CE-NET, Concurrent Engineering Network of Excellence, ESPRIT project (No25946) during 1997–1999, and in Beyond the GUI, User-Centred Intelligence Eureka/ITEA project (No: Eureka 2023 / ITEA 99002) from 1999–2001.

I have worked in several national and European research projects during my career, and I must confess that the drive, atmosphere and level of co-operation that I experienced in VIRPI was exceptional. I wish to express my highest gratitude to all of the researchers in the VIRPI project team. I would especially like to thank Professor Petri Pulli, and Mr. Marko Salmela from VTT and Dr. Tuomo Tuikka from the University of Oulu for their supporting work and innovative, but critical discussions related to the topic. My warm thanks also go to the other researchers in the project at that time including Mr. Tommi Anttila, Mr. Harri Kyllönen and Mr. Peter Antoniac (now Dr.) at VTT, Professor Kari Kuutti and Professor Jouni Similä at the University of Oulu, Professor Turkka Keinonen and Dr. Simo Säde at the University of Art and Design Helsinki, and Mr. Petri Kotro at the University of Lapland.

I am also grateful to the industrial members of the VIRPI project – Nokia Mobile Phones, CCC Group, JP Metsävainio Design Ltd. C3 Suunnittelu Ltd., Polar Electro Ltd., Benefon Ltd. and Elektrobit Ltd. Not all of these companies exist in the same form anymore. There have been mergers and other reorganizations of the companies. However, the memories and results from the co-operation remain, and therefore I wish to express my warm thanks to the representatives of those companies: Mr. Timo Tokkonen, Mr. Keijo Kuosmanen, Mr. Pekka Pärnänen, Mr. Jukka-Pekka Metsävainio, Mr. Ari-Pekka Töykkälä, and Mrs. Ira Laitakari. The comments and case studies provided by you gave real value for my work.

I reactivated my thesis in 2004 at Cybelius Software Ltd. after three years of work in a very interesting company with challenging products and international customer contacts. At Cybelius Software I was lucky to be able to continue my work in an exceptional team. I was able to work further with Mr. Marko Salmela and Dr. Tuomo Tuikka, who also joined the company. In addition, I met many new professionals. I wish to express my warm thanks to all the Cybelius personnel for their fruitful co-operation and sharing both the moments of success and failures within the hard-working environment of the company. Special thanks go to Managing Director Jyri Pyrrö, who encouraged me to continue the thesis and helped me in organizing the working periods for the finalization of the thesis.

My work at Cybelius was related to building business for a new software product, Cybelius Maestro. In this position I had the opportunity to visit and experience many world-class companies in Europe, Japan, Korea and the USA operating in the area of electronics and telecommunications. I met many interesting persons in these companies, not only as highly professional experts, but also as close individuals sharing the joint moments of work and private life. I wish to thank the companies Nokia and Polar Electro, which were involved in the research, and especially their representatives Mr. Timo Tokkonen, Mr. Antti Kumpula, Dr. Heikki Huomo and Mr. Ilkka Heikkilä for their fruitful co-operation. There are also many other individuals from different companies that have worked closely with me during the Cybelius period. I am very sorry that I cannot mention your name publicly because of our confidential business relationship. Nevertheless, I am sure you recognize yourselves as important contributors to my work. Thank you all for your support of my work!

In May 2005 I was pleased to agree with VTT on the finalization of my thesis at VTT for the VTT Publications series. Without these arrangements it would have been very difficult for me to complete my research process. I am really grateful to research manager Dr. Hannu Honka and Dr. Tua Huomo from VTT for helping me in having this possibility to work at VTT again for a short period. I also would like to thank all of the other 'hidden' people at VTT that continually support us, the researchers, in our research work. Warm thanks also to Mr. Seppo Keränen from Wordsmith for proofreading my thesis.

I am grateful to Professor Juha Röning for supervising my thesis. I also thank Professor Veikko Seppänen and Dr. Kari Leppälä as unofficial advisors and supporters of the work. Despite my long research schedule, you all have always been ready to comment on my work, and support and motivate me in continuing towards the final goal.

I would like to express my gratitude to Professor Kulwant Pawar from the University of Nottingham and Dr. Pekka Ketola from Nokia as the pre-examiners of this thesis. Your steering and comments regarding especially the research methods, overall structure and contribution of my work were really valuable. You guidance had a significant role in improving my work as a scientific thesis.

The financial support provided by VTT and Tekes is gratefully acknowledged, and the role of the Federation of Finnish Metal, Engineering and Electrotechnical Industries, FIMET, as a backing force behind the VIRPI project, is greatly appreciated. I would also like to thank the Seppo Säynäjäkangas Science Foundation for supporting my work.

Finally, I wish to express my warmest thanks to my wife Soile, my son Oskari and my daughter Susanna, for their support and understanding throughout my work. Finalizing this thesis in addition to a normal full-time job and family responsibilities has not always been easy to juggle. I apologize for my occasional absence from normal family life, and will compensate for these periods in the time to come!

Kempele, Finland, August 2006

Mikko Kerttula

Contents

List of symbols

2D	2-dimensional
3D	3-dimensional
API	Application Programming Interface
ASIC	Application-Specific Integrated Circuit
C++	An object-oriented programming language
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAEE	Computer-Aided Electronic Engineering
CAM	Computer-Aided Manufacturing
CASE	Computer-Aided Software Engineering
CDC	Connected Device Configuration, a specification framework for Java applications that can be shared across a range of network-connected consumer and embedded devices, including smart communicators, high-end personal digital assistants (PDAs), and set-top boxes.
CE	Concurrent Engineering
CLDC	Connected Limited Device Configuration, specification framework for Java applications targeted at devices with very limited resources such as pagers and mobile phones.
CSCW	Computer-Supported Co-operative Work
CVE	Collaborative Virtual Environment
DSP	Digital Signal Processing
EAI	External Authoring Interface
E-business	Electronic business
E-commerce	Electronic commerce
EDA	Electronic Design Automation

EP	Cybelius Maestro Extension Package
FIMET	Federation of Finnish Metal, Engineering and Electrotechnical Industries
FPGA	Field Programmable Gate Array
GSM	Global Systems for Mobile communications, a mobile phone system
GUI	Graphical User Interface
IC	Integrated Circuit
IDE	Integrated Development Environment
IT	Information Technology
Java	An object-oriented programming language developed by Sun Microsystems
Java3D	3D graphics API for the Java platform developed by Sun Microsystems
Java ME	Java Micro Edition. Java platform for resource constrained devices such as PDAs, cell phones and other consumer appliances.
LAN	Local Area Network
Maestro	Cybelius Maestro
ODM	Original Design Manufacturer
OEM	Original Equipment Manufacturer
OpenGL	A standard software interface to graphics hardware developed by SGI
PCB	Printed Circuit Board
PDA	Personal Digital Assistant
PDM	Product Data Management
PPP	Phased Project Planning (NASA process model)
PUI	Physical User Interface
QFD	Quality Function Deployment

R&D	Research and Development
RAD	Rapid Application Development
RF	Radio Frequency
ROOM	Real-Time Object-Oriented Modelling
SDK	Software Development Toolkit
SGI	Silicon Graphics Inc.
SME	Small and Medium size Enterprise
SoC	System-On-Chip
TAPI	Telephone Application Programming Interface
TCP/IP	Transmission Control Protocol/Internet Protocol
Tekes	The Finnish Funding Agency for Technology and Innovation
TQM	Total Quality Management
UCD	User-Centred Design
UDL	Universal Description Language
UI	User Interface
UML	Unified Modelling Language
VAS	Value-Added Service
VIRPI	A research project: Virtual Prototyping Services for the Electronics and Telecommunications Industries
VHDL	Very High speed integrated hardware Description Language
VR	Virtual Reality
VRML	Virtual Reality Modelling Language
VRP	Virtual Reality Prototyping
WLAN	Wireless Local Area Network
WWW	World Wide Web

1. Introduction

Globalization and evolving technologies have set new demands on companies and organizations. Companies applying high technology solutions or operating in the area of consumer and entertainment electronics must in particular be able to constantly react to changing business circumstances. Tightening global competition, distributed organizations, advancing technology, shortened product life cycles, rapidly changing user preferences, and increasing product complexity all require effective, new methods and practices to be utilized in product development, manufacturing and marketing.

Product life cycles have become rather short especially in tightly competitive product branches of the high tech industry. With products such as software, computers and mobile phones, this development is mainly due to expanding markets and technological progress. The fast evolving nature of high technology markets and the enormous business potential it offers for companies are addressed by Moore (1999), for example. New products or product releases with new features offer the possibility for a company to excel beyond other manufacturers, and as such they are launched frequently in order to ensure or improve their current market position in the ongoing competition. It is also true that new product releases often have better returns – it is easier to sell and set the price for state-of-the-art products.

For customers, new products are expected to offer more benefits and user satisfaction. Technological advances can help companies in providing these benefits. However, because of the competition, technological differences between the companies in many areas of the electronics industry have become rather small. In addition, there are application areas in which the significant part of the technology is regulated by laws and standards. For example, in the telecommunications industry the core technology for third generation mobile systems is decided together with international standardization organizations and major technology providers (3GPP 1998). These common regulations and standards narrow the area of technological competition, at least in certain technology areas, and shift the focus of the competition to other issues.

Due to the previous facts the impact of pure technological advantage in business could be said to have decreased during the 1990s. Of course, there are and will be product features that are based on technology advances of the company and that offer competitive advantages for some period of time. But, it is increasingly expected that this advantage will be caught up with by competitors rather rapidly. For example, Moore (1999) mentions a dominant design effect of technological solutions tending to focus rapidly among companies, which decreases real competition. With many products, it is not a specific technological feature any more that will guarantee a competitive advantage, but more often success is based on other product attributes such as the selected feature set, total quality, design and usability of the product. Besides for these product specific features, the company's skills in different business operations and strategies, such as in product variation, marketing and sales, R&D, product development, manufacturing and logistics, play a major role in overall business success.

1.1 Personal electronic products

Personal electronics is a loose definition for electronic devices that people use for their everyday tasks. Basically, it can include everything from wrist computers and Personal Digital Assistants (PDA) to PCs and mobile phones. The definition can overlap with many other common product categories of portable, wireless, consumer and entertainment electronics. As there is no commonly agreed upon definition for a personal electronic device, for the purpose of this thesis, it is defined with the following characteristics. Personal electronics include products that:

- offer personal services for its user;
- include a user interface with screen and software controlled operations;
- are customisable or personalisable for the users' needs;
- are not limited to stand-alone devices, but devices that need an infrastructure offering operating environment and services for them[1]; and

[1] This characteristic adds an interesting 'system design view' to the development of personal electronics. It is thus not anymore a single device that has to be taken into consideration in the development, but a system that may consists of several devices, services and sub-systems communicating with each other.

- have limitations in one or more of the following resources: computer power, size and/or power consumption.

Personal electronics are not limited to portable or wireless electronics, as consumer electronic devices such as a set-top box or home theatre system are also accepted into the category. Personalization is not either restricted to a single user use because a group of users, like family for home electronics, can control the same device, but with personalized features. It must also be noted that a PC or laptop are not considered as personal electronic devices in this thesis. This exclusion is based on the last item in the above list.

The given definition of personal electronics is close to that of information appliances, which are defined by the National Institute of Standards and Technology, USA, (NIST 1994), Norman (1998) and Mohageg and Wagner (2000)[2]. NIST suggested the term information appliances for systems that support communication, information storage, and user interactions. Norman considered information appliances as simple devices that are targeted towards limited tasks, but have the ability to be connected to each other, and as such form a general device service network around the user. There are two important issues that separate personal electronics as defined above from information appliances. Personal electronic products are not necessary so restricted in services and operations. They also include products such as smart phones and PDAs that are not meant to be used any more only for single limited tasks. Additionally, especially in their early definitions, information appliances are not considered as personalized devices.

Personal electronic products also include many features from smart products as defined by Keinonen et al. (1996). The authors emphasize the role of dense user interface (UI) in a design product. Since the definitions for smart products also vary a lot, as summarized in (Säde 1999), and these definitions do not generally recognize the personalization aspect of the product, the term personal electronics was considered better suited for the purposes of this thesis.

---

[2] Additional definitions for information appliances are available e.g. in http://www.usabilityfirst.com/glossary/term_846.txl and http://www.linuxdevices.com /articles/AT6664732218.

1.2 Virtual design and virtual prototypes

As stated above, the requirements for a successful development of personal electronic devices are diverse. One approach used to meet these requirements is *virtual design* and *virtual prototypes*. Virtual design introduces a product and business development framework that is based on effective use of realistic and functional product simulations called virtual prototypes. The idea is to incrementally and iteratively build the virtual product prototype and benefit from it through the entire product life cycle. The virtual prototype and its parts evolve from concept design to product specification, implementation, marketing and after-sales. Additionally, virtual prototypes can be employed in usability testing, product localization and documentation.

Virtual design aims to improve the communication between product team members, to enhance validation of product concepts via advanced simulations, to increase the efficiency of component and material reuse in different life cycle phases, to reduce simulation costs, and to promote concurrency in design and business tasks. These aims are further summed up in important company benefits, such as shorter product development cycles and time-to-market, reduced development costs, and also improved accuracy and quality of development work, in order to meet the needs of the customer and market.

Virtual design is applicable in many diverse engineering tasks in local company operations. However, it also has a lot of potential especially for multisite companies using modern Internet and IT-technologies. The Web, for example, provides interesting opportunities to benefit from virtual prototypes in, e.g., collecting customer preferences, distributed product development, electronic commerce, web-based marketing, training, and service and maintenance.

Why are virtual design and virtual prototypes currently interesting? Changing business environments as discussed above are driving companies to look for new, more efficient tools and methods. Along with the development of computer and software technologies, they are also studying virtual design and virtual prototypes as a promising approach for future challenges. It is also obvious that many other methods and terms activated in the 1990s, such as CE (Concurrent Engineering), virtual organizations, distributed design, prototyping approaches, are favouring virtual design because they share many common goals and practices. From a technical point of view, a virtual prototype offers promising benefits especially for the development and business of software-centred products. It is one of the first simulation technologies that has its focus on software features, but that can still extend its focus to other needed technologies in order to create a complete simulation experience for a modern personal electronic product.

1.3 Research approach

In this thesis, the experiences gained with virtual design and virtual prototypes in the development of personal electronic products are presented and analyzed. The thesis represents a constructive research in the field of technology and design science. In addition, as is typical of the constructive research approach, the study implements the qualitative research approach using observations, their analysis and generalization to deduce the final research results and conclusions.

There is no exact definition for technology, but it is typically understood to consist of artefacts, processes and methods for creating these artefacts, and of knowledge and information related to these. Similarly, the definition of technology as a research field or science is ambiguous. For example, Simon (1969)[3] has described technology and design related research as a science of artificial as opposed to sciences of natural, as it produces new artefacts and knowledge by applying natural sciences and scientific methods. March and Smith (1995) complement this view by stating: "*Whereas natural science tries to understand reality, design science attempts to create things that serve human purposes*." They further state that the outcome of the design science is assessed against the questions: does it work, or is it an improvement? In this thesis the goal is to study if virtual design can be implemented and if it can be beneficial for companies.

---

[3] Simon A. Herbert, a winner of the 1978 Nobel Prize in Economics, wrote his book "The Sciences of the Artificial" already in 1969, and has updated it in 2 additional editions in 1981 and 1996.

In practice, the research in technology and design science can be divided into the following three main categories[4]:

1. Research on artefacts and their features; how to solve technical questions, and how to improve the performance of artefacts?
2. Research on the methods, tools, systems and processes used for designing, developing and manufacturing artefacts; how to operate more efficiently, and how to reduce time and effort in the operations?
3. Research on the interactions and influence of technology; how technology is used, utilized and adopted, how technology impacts on and interfaces with people and other domains?

According to this classification, this thesis falls into the second category, and its coverage can be further illustrated by a company model given in Figure 1. As stated in the figure, the main focus of this work is on product development processes, and technologies and methods. This dualism is a dominant characteristic of the entire thesis, as virtual design (cf. process approach) and its supporting tools (cf. technology and methods) are presented parallel throughout the text. Organization related issues have also been addressed during the work, but a focused organizational view is left out of the thesis in order to avoid too extensive a research work. In fact, there are numerous ongoing research activities in the area of virtual organizations and enterprises, which have a good potential also to cover the organization aspects related to virtual design.

---

[4] This practical categorisation follows the ideas already presented by Simon (1969), when he mentioned engineering, systems and management sciences as the main categories of the research in technology and design. These sciences bear a clear correspondence to the numbered categories, especially when the category of management science is extended with organizational, social and cultural sciences and economics in general.

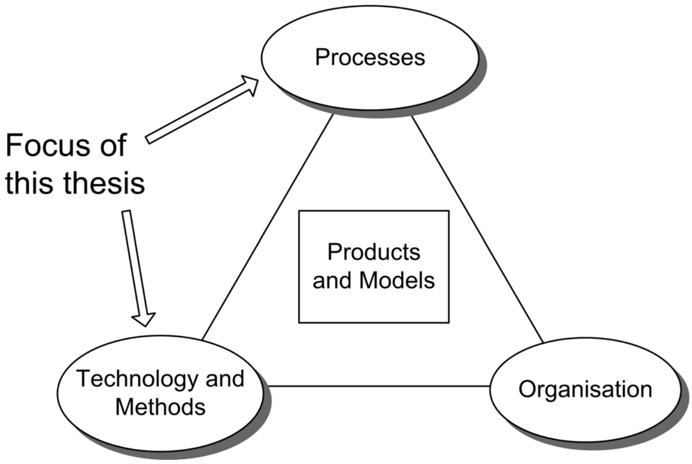


Figure 1. Focus of the thesis.

In view of Figure 1 and the overall scope of this thesis, it is worth noting that in this research product development is defined to include also usability engineering, concept design and product specification – operations that are not regarded as a part of product development in all organizations. With this broader approach, the role of product development is emphasized as a significant part of the business in a high technology company. The adopted definition extends the role of product development from a passive service to an active and co-operative organizational activity, which has a major impact on the overall success of a company.

The research reported in this thesis also represents the applied research approach. In contrast to basic research, which targets at understanding and gaining knowledge of fundamental aspects of phenomena and facts, applied research is targeted at solving specific, practical questions. March and Smith (1995) state that the distinction between basic and applied research boils down to the question how closely the research impinges on practice. The authors further state that research in design is usually categorized as applied research and research in natural science as basic research. Even though this rule is not definitive as such (March and Smith 1995), it applies here as this thesis reports research that aims to solve real problems faced in product development.

1.3.1 Definition of the research problem

The goal of this thesis is to find out, if and how virtual design can help companies develop personal electronic products. The motivation for the research comes from the practical findings and problems experienced by the author in different companies and co-operation projects. There are challenges in developing personal electronic products effectively in companies. In practice, they mainly relate to the development process model that is applied, the used tools and the understanding of the role of good prototyping and simulation practices.

To find out the potential of virtual design, the problem areas of the current practices must be specified and the requirements for improved practices must be defined. After the model of virtual design is defined, it will be evaluated against the requirements. An analysis of the performance of virtual design could only be performed theoretically by linking the parts and execution model of the virtual design process to the requirements. However, as an applied research, the thesis aims to provide practical guidance and suggestions. This is why the research is continued with case demonstrations linking the performance of the presented process model to actual design studies. These cases are presented not only to prove the possibilities of virtual design, but also to provide practical guidance as to how it can be implemented in companies.

The research problem for this thesis is the following:

> How can virtual design, a product development process framework utilizing virtual prototypes, be used effectively in the development of personal electronic products?

The problem can be further divided into the following sub-problems:

1. What are the limitations of the typical product development process models and tools used in electronics industry, when they are applied within the development of personal electronic products?

2. What are the requirements for the product development process model and development tools supporting the development of personal electronic products?
3. How can the virtual design process model (as defined in Chapter 5.5) utilising virtual prototypes (as defined in Chapter 5.1) meet the requirements set for the product development process model and development tools supporting the development of personal electronic products?
4. How can the virtual design process model utilizing virtual prototypes be taken into use and implemented in a company?

These research questions are answered in Chapter 8.2.

1.3.2 Research method and techniques

The research method applied in this thesis represents the constructive research approach. This method is widely used in technology research and design sciences. For example, Iivari (1991) reports constructive research, including conceptual and technical development, as one of the three main research methods for information systems[5]. Similarly, March and Smith (1995), and Järvinen (1999) introduce constructive research as a typical research method for design sciences. Järvinen (1999) links constructive research to research that stresses the utility of innovations[6]. A classification of research methods based on Järvinen's taxonomy is illustrated in Figure 2. March and Smith (1995), and Järvinen (1999) divide constructive research to the two main phases of construction and evaluation. However, these phases are usually preceded with analysis phase that focuses on studying problems that the construction aims to solve, and forming requirements for its implementation and evaluation. Similarly, in this thesis the main research phases are analysis, construction and evaluation.

---

[5] The two other are nomothetic, which covers formal-mathematic analysis, experiments and field studies, and idiographic, which includes case and action research.

[6] Järvinen (1999) has selected the term innovation instead of artefact so that the coverage of the construction is not limited to technical aspects but can include also social, informative and theoretical aspects of innovations.

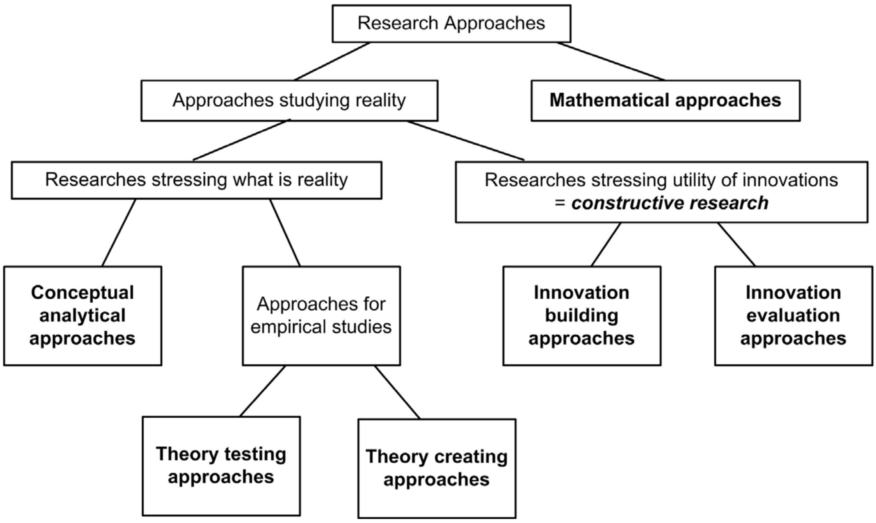


Figure 2. Classification of research methods (modification from Järvinen 1999).

The constructive research method provides a research framework with the named main phases, and allows different research techniques and procedures to be used within each main phase[7]. In this thesis, the practical research work is focused on empirical research using observations and experience both in the analysis of the problem areas in product development and in the practical evaluation of the introduced virtual design framework in the industry. Empirical research is generally understood as research that improves our understanding of the empirical reality around us. Empirical research uses data from observations and experience to address research problems and questions. Lauer and Asher (1988, p. 7), for example, define empirical research as "*the process of developing systematized knowledge gained from observations that are formulated to support insights and generalizations about the phenomena under study*". Potts (1993) also speaks for empirical research, especially in the area of software engineering; the author highlights its benefits of the so-called industry-as-laboratory methodology compared to the research-then-transfer methodology. He defines 'industry-as-laboratory' as an approach in which "*researchers*

---

[7] Järvinen (1999) also discusses whether or not constructive research could be considered as a so-called super-method that applies other research methods within its phases. The super-method approach was originally proposed by Nunamaker et al. (1991) based on their experiences in information system design.

identify problems through close involvement with industrial projects, and create and evaluate solutions in an almost indivisible research activity" (Potts 1993)[8].

As a constructive research, this thesis focuses on the development of construction, i.e. a virtual design process model, and its performance evaluation against given criteria. These phases of construction and evaluation are preceded with an analysis phase that studies the general challenges met in the electronics and telecommunications industry and defines requirements for better working practices. The three main research phases and their tasks are described below:

A. Analysis phase

A1. Study the current practices in the industry. In this phase information regarding tools, processes and practices in product design and system engineering was gathered from articles, books and personal contacts. The coverage of design and application areas in this phase was kept broad because the objective was to gain a general overview of adopted practices in companies developing software-based products or systems with the help of modern information technology. Most of the material was gathered through the author's experience between 1996 and 2001 in various contract and joint research projects at VTT, as well as through customer contacts at Cybelius Software (later Cybelius) since 2001. This task relates to Chapters 2 and 3 of the thesis.

A2. Analyse the problems encountered in the development of electronic products. This phase shifts the focus of the work to its target application area. The goal is to identify the problems and deficiencies of current practices in electronics design with regard to personal electronics products. The work here is based on design cases, which are described in more detail in Chapter 1.3.4 below. The results of this phase are presented in Chapter 4.1.

---

[8] The three main changes Potts (1993) sees when adopting the industry-as-laboratory approach are: (1) It defines problems empirically, which makes them more reliable. (2) It emphasizes a case study, e.g. a real system-development project, that "becomes a way to obtain knowledge and appreciate its significance", and does not only demonstrate research results, but produces them. (3) It puts a greater emphasis on contextual issues, i.e., extends the research to recognize also the non-technical factors, that cannot be recognized in laboratory research.

A3. Perform a requirements analysis for a new product development framework and supporting design tools. The requirements for an enhanced product development process model and development tools are defined. The task is conducted by analyzing the results of previous steps, and by taking into account direct feedback from design cases. The requirements are documented in Chapter 4.

B. Construction phase

B1. Define and construct virtual design and virtual prototypes for new product development practices. Virtual design and virtual prototypes, i.e. innovations, are represented as promising approaches to fulfilling the requirements given in the evaluation phase. This task refers to Chapter 5 of the thesis.

B2. Implement and evaluate design studies. The introduced virtual design model is partially implemented and executed through selected public or industrial demonstrations. Prior to moving on to case demonstrations, the tool support for virtual design is analyzed and two tool platforms are introduced to be used in design studies. An evaluation of the demonstration cases is performed against the requirements presented in task A3. This phase is documented in Chapter 7.

C. Evaluation phase

C1. Validate the construction, i.e. the concept and model of virtual design. This research task analyses the presented design studies as a whole and validates the combined results against the research problem and current practices that are presented in task A2. The performance assessment of the construction is partially done through the experiments and partially derived from previous experiments. The validation phase is presented in Chapter 8.

C2. Discuss the contribution of the research. The evaluation of the research as a whole is done based on the general evaluation criteria set for constructive research. Here the novelty of the innovation and its performance are in the main role. Additionally, as the construct of virtual design represents a method that can be classified based on its research outputs according to the ideas presented by March and Smith (1995), and

Järvinen (1999), it is evaluated against the criteria set for its class. This phase is presented in Chapter 8.3.

1.3.3 Motivation for the selected research method

When the research plan for this thesis was created, it was considered important that the research and its results should have a close connection to real and practical work in industry, and that they should also aim to propose improvements. When selecting the research methods, this goal led naturally to methods that focused on studying real world issues (Figure 2) and that emphasized the applied empirical research aspects. These basic assumptions left out, for instance, conceptual analytical research methods, as these focus only on describing existing practices and not on creating new ones. In addition, mathematical theories were not considered practical as in view of describing the research to target audience including practitioners from industry. Practitioners rather like to see the results in a practical form using the same terminology, questions and answers they are using in their every-day work.

Taking into the account the above requirements for the research method, there were still some other alternatives in addition to the constructive research approach. Among the theory creating approaches, especially the case study (Yin 1994) and ethnographic (Van Maanen 1979) methods could have been used in creating a theory of virtual design. However, these methods were not seen as promising choices in practice because they did not emphasize the value of new constructions, i.e. the innovation of virtual design. Additionally, due to the unpredictability of co-operation with the companies that were involved in the research, it would have been impossible to make any detailed implementation plans needed for case studies, or to agree on any sort of long-term residential research as used in ethnographic research.

Action research (Hult and Lennung 1980) is another typical research method applied in the field of technology and design science. In this approach, the researcher acts as an active player in a collaborative team working on a real problem case. The researcher makes observations about problems, suggests improvements and evaluates these as a member of the team in a cyclical process. The goal is to improve an existing artefact, system or process parallel to its study. Action research could have been applied also in this thesis, if it had been possible to agree on and plan the needed arrangements with some of the case companies. However, due to the extensive focus of this research, i.e. virtual design as a product development framework, and also owing to its novelty, it was impossible to tie companies to this kind of co-operation. Consequently, the constructive research method relying on collecting evaluation results from several design studies and making deductions on the basis of these was considered a more suitable approach and selected as the main research method for this study.

1.3.4 Design studies

The ideas and methods set out in this thesis are based on a close co-operation with companies designing, developing, manufacturing and selling personal electronic products. Although the companies concerned represent both large organizations and small and medium-sized enterprises (SME), they all share the features of emphasizing the role of software development practices, using advanced software and hardware technology, and showing continuous awareness of changing customer preferences. In this thesis, the research is focused on the above assumptions.

The development of personal electronic products was studied from the viewpoints of both manufacturer companies and their subcontractors. The product ranges varied from home electronics to hand-held personal devices. As a common feature, all the products involved developing mechanics, electronics, software and wireless telecommunication components. Industrial design aiming at appealing appearance and UI development supporting good usability were among the major requirements set for the products. On organizational level, the companies represented international organizations with several offices and wide distribution channels. This global operation had an effect both on products (e.g. different language versions, local regulations and laws, et cetera) and on working practices (e.g. distributed and collaborative design through different time zones).

The design studies in industry were conducted in two phases. The main phase was conducted between 1996 and 2001 in contract and joint research projects at VTT. The results were updated by the second phase between 2001 and 2005 at

Cybelius Software. The latter period provided some highly significant updates to the thesis research, especially in the context of international business, as during this period several world-leading companies in Europe, Japan, Korea and the USA were included into the study. The actions with these companies included meetings and discussions with different design and business teams, along with joint evaluation cases and actual business cases in connection with the Cybelius Maestro product.

Overall, multiple cases were studied, while not all of the actual names of case companies cannot be mentioned due to confidentiality reasons. This means that in most cases the companies are safeguarded with anonymity and generalizations of business information. However, to concretize, at least at some level, the business areas and geographical locations of the companies that the author has been co-operating with and studying during the research work are summarized in Table 1. The company categories defined in the table, e.g., ME, OE or HJ, are used for linking the comments and results of subsequent chapters to these specific company types. Note that the designation 'location' in the table means the location of the company unit or people that were co-operating. The company itself may have also other international operations in addition to the referred location.

The results of the industrial design studies are summarized mainly in Chapters 4 and 7.3, but also Chapter 8 "Analysis and Discussions" includes some references to case experiences. Additionally, two further public design studies that support industrial design studies are introduced in Chapter 7.

Table 1. Summary of companies contributing to the design cases.

	Location			
Business area	**Europe (E)**	**Asia (A)**	**Japan (J)**	**USA (U)**
Mobile communication (M)	ME 2 companies	MA 4 companies	MJ 3 companies	MU 3 companies
Home and office electronics (H)	HE 1 company		HJ 1 company	HU 2 companies
Other personal electronics (P)	PE 2 companies			
Professional electronics (Pr)	PrE 2 companies			
OEM and ODM (O)	OE 2 companies			
Design companies (D)	DE 2 companies		DJ 1 company	

1.3.5 Author's research history

The motivation for this thesis derives from the challenges the author has met in his career both in companies and at VTT Technical Research Centre of Finland. Between 1990 and 1995, the author worked at Rautaruukki New Technology, which developed and sold machine vision based surface inspection systems for flat steel products. Already during that time, he recognized that there were a lot of challenges in managing the co-operation and communication in a multi-discipline development and project organization.

In 1996 the author joined VTT[9] and started his work as a project manager in the VIRPI[10] project. The project included VTT, University of Oulu, University of Art and Design Helsinki and University of Lapland as research participants and Nokia Mobile Phones, CCC Group, JP Metsävainio Design Ltd. C3 Suunnittelu Ltd., Polar Electro Ltd., Benefon Ltd. and Elektrobit Ltd. as industrial partners. The aim of the project was to study and demonstrate the possibilities of virtual prototyping in the development of hand-held electronic products. During the project, the author was responsible for managing the ComPen public pilot case. In addition, he participated in virtual prototyping technology transfer and requirements analysis projects at Nokia Mobile Phones. Later in 1998 he also had a working period in Cybelius Software Ltd. and CCC Group, where he participated in product definition and development work, and also in beta-customer projects.

Already in 1996 the author contributed to introductive articles on virtual prototyping (Pulli el al. 1996, Kerttula and Similä 1996, Kerttula et al. 1996). In the VIRPI project marketing and e-commerce were also researched as application areas for virtual prototypes, and were reported in some articles. In 1997, new articles were contributed to. In (Similä et al. 1997) virtual prototypes were studied in the context of Internet marketing and product customization. In (Kuutti et al. 1997) some important basic ideas of virtual design and virtual

---

[9] The research unit at VTT at that time was VTT Electronics in the city of Oulu.

[10] VIRPI, Virtual Prototyping Services for the Electronics and Telecommunications Industries, was a national research project funded by Tekes, the Finnish Funding Agency for Technology and Innovation, and industrial and research partners, 1996–1999.

design space were introduced from the communication and co-operation point of view. ComPen, the public pilot case of the VIRPI project, was introduced in (Kerttula et al. 1997a) and in (Kerttula et al. 1997c). The idea of virtual design process and the initial version of the process model presented in this thesis was introduced in (Kerttula et al. 1997b). In 1999 additional papers were published (Tuikka et al. 1999, Kerttula and Palo 1999). In (Kerttula et al. 1999b) the possibilities and challenges of using virtual prototypes in web-enabled engineering were described. In 1999 the final report of the VIRPI project was also published (Kerttula et al. 1999a).

The VTT projects presented new research focusing on the utilization of virtual prototyping technology in the development of hand-held electronic products. The research target was wide because it aimed at a seamless use of virtual prototypes not only in product development, but also in other business tasks, especially, in marketing and sales. Because of its specific focus, the research conducted was unique also on a global scale. The author started to write his thesis in 1999. In the first versions, the focus of the thesis was predominantly on web-enabled virtual prototyping and distributed virtual design. In 2000 the author together with Mr. Tokkonen from Nokia initiated the negotiations for the commercialization of the VRP tools, which were developed within the VTT strategic Virtual Reality Prototyping (VRP) project and further customized in the VIPRI project. The negotiations resulted in the IPR being transferred from VTT to Cybelius Software. In May 2001, the author joined Cybelius Software as a business development manager.

An important summative paper on virtual design was released by IEEE Computer in November 2001 (Kerttula and Tokkonen 2001). The paper introduced the virtual design process for the development of multi-engineering electronics systems and reported its case study at Nokia Mobile Phones. In 2001 the author also contributed to two papers introducing the possibilities of virtual prototypes as a front-end of collaborative PDM (Salmela et al. 2001a, Salmela et al. 2001b). In 2002 the author presented one working paper in Wireless World Research Forum on applying virtual prototyping in ensuring the positive user experience of wireless products (Kerttula and Tokkonen 2002). He also summarized the virtual design approach and Cybelius Maestro as a supporting tool for the approach in a book released in 2003 (Leppälä et al. 2003). The author was the main contributor for the second part of the book.

The author continued working on his thesis at Cybelius Software[11]. There were some short updating periods for the material between 2001 and 2003, but the main work was performed during the period starting at the end of 2004 and ending in 2005. The focus of the thesis was narrowed, by leaving web-enabled issues out of the work, and by concentrating on personal electronic products. Without this re-focusing the thesis would have become too broad to manage.

A number of interesting customer cases applying, at least, partly the principles of virtual design were implemented at Cybelius Software between 2001 and 2005. From the adopted research perspective, the problem in these cases was that they were mostly strictly confidential and thus not available to the research community for any detailed discussion or description. A valuable exception in these cases was that in 2005 Cybelius prepared a joint IEEE paper with Nokia on using virtual prototypes in the concepting and specification of new mobile services (Repo et al. 2005). The author contributed his experiences of virtual design and virtual prototyping to this paper.

This thesis reports virtual design and its application in personal electronics. Besides this main track it also reports the research history of the Cybelius Maestro products. The Maestro tools have evolved from strong and innovative national research co-operation. There are not many similar cases, at least in Finland, in which the origins of software products and also the company can be tracked so clearly to specific research projects and co-operation. The author recognizes the history of Maestro tools and his role in the evolution of and in current Maestro business. Maestro tools have been reported in this thesis. However, the goal is not to promote them as commercial tools, but as an enabler of virtual design. Without presenting Maestro the content and the contribution of the thesis would have been incomplete.

---

[11] Cybelius Software Ltd. was integrated to CCC Group on 1st July 2005. Cybelius Software still remains as an auxiliary business name and the development and sales of the Cybelius Maestro product are continued in this new organization.

2. Challenges in the electronics industry

This chapter provides a short introduction to the challenges that are faced in the business of electronic products at present. The study was performed especially from the view point of global business targeting mass customized but yet personalized electronic devices. The goal is to concretize the significant changes that the industry has witnessed since the beginning of the 1990s. It is clear that these changes place pressure on companies to look for new business and R&D practices.

2.1 The dominance of software

The use of software has increased in many different products. Even the more traditional products, which used to be based on pure mechanical or electronics parts, today increasingly include software components, not only in their embedded control core, but also in user interfaces. This progress has meant that many companies that previously concentrated on more traditional products, have also been increasingly tied to software technologies and development practices. The role of software has also increased dramatically in electronic products since the 1990s. For example, in mobile phones, the size of software has risen from some kilobytes in analogue phones to several megabytes in the latest smart phones. An interesting feature in this progress is that the weighting of the software is shifting from lower level system software to the UI parts. For example, in some consumer products UI software can account for over 50% of the entire software. Software engineering has a major role in the development of these products.

Software development is a demanding task. This fact has been noticed, for example, in many companies changing from traditional industries of mechanics and manufacturing to software engineering. Many managers and experts with experience in the design of concrete parts and products thought that software can do everything. “You just programme it.” However, in time it was noticed that it was especially software related project tasks that were likely to be late. It was also noticed that small programs that were developed and updated by a small group of people, or even by a single expert in a company, usually worked fine, but when the size of the program and project team grew, problems would arise easily. It was also quite surprising that in software projects a simple increase in effort could not speed-up the project. This special nature of software engineering was observed by Brooks (1975). He wrote his historical book “The Mythical Man-Month” as early as 1975, and even today high quality software production is a demanding goal to achieve.

Software and software development are something different compared to traditional engineering areas, such as civil, mechanical and hardware engineering. Software is not constrained by the laws and characteristics of the physical world, which makes software difficult to imagine. It is not as tangible to build as other product parts are. Software parts, i.e. programs, functions, code, design documents, et cetera, and their contribution to the complete product assembly is difficult to see or experience in advance. This fact sets strict requirements for design. Even though a product or software architecture should be designed as well as possible before the implementation, usually several iterations are needed to correct and fine-tune the design during the implementation. Often iterations are also done because software is easy to change. This characteristic makes software, on the other hand, a very interesting, but also a very difficult design area. You can make modifications and even a new product version easily by making changes to the existing code, but you can also easily ruin the complex design of the software. Additionally, software is difficult to measure, and because it is still a relatively young design area, its development techniques, tools and target environments are still changing fast.

Despite its challenges in design and implementation, software offers interesting possibilities for companies. When implemented successfully, software can be copied inexpensively. The possibility to attain substantial profit with relatively small investments exists. In addition, because of its easy modification, it offers a cost-effective way for product versioning and customization. In fact, software has become a critical factor in the electronics business as an enabling technology and product feature.

2.2 From technology to user satisfaction

Implementing a new product or product features as such may not be a problem, because technological progress offers new ideas and solutions. Rather, the problem is often how to form an optimal product variation for different customer segments in international markets. The trends and preferences in consumer markets are very difficult to envision – they differ among customer segments and may change quickly.

One common trend of the 90's business has been the aim at increasing user involvement. The goal has been to bring more benefit and satisfaction for the customer. In industrial and consumer electronics, this customer-centred business approach has been recognized as an effective method, and while it was first only implemented in marketing slogans and written company values, today it is finally also applied in real everyday operations. In practice, the driving force in this progress has been the investments in industrial design and usability engineering.

In the 1990s, the number of industrial designers was increasing in companies in which industrial design was acknowledged as an important part of the product development process. On the one hand, the value of product design has been recognized not only in the area of consumer products, but also in original equipment manufacturer (OEM) and original design manufacturer (ODM) businesses and industrial electronics. Industrial design, on the other hand, has a strong connection to usability. In fact, good industrial design should support good usability, which, as defined by ISO/DIS 9241-11 (1998) refers to the effectiveness, efficiency and satisfaction of accomplishing a specific task in a specified context of product use. In practice, product usability is a complex dimension made up of various features such as its looks, form, weight, size, display, keys, sounds, functions, menus and services. To find the right balance between these features, companies have established new departments and laboratories for usability engineering. The goal is to improve their knowledge in user-centred design (UCD) as defined in the ISO 13407 standard (1999).

2.3 From technology management to knowledge management and effective co-operation

The development of an electronic or telecommunication product is a typical multi-technology process incorporating several engineering disciplines such as software, hardware and mechanical engineering. Each of these design domains has its own modelling conventions and tools. For instance, software designers may use object-oriented or real-time modelling notations for design, C/C++ code for implementing the software and special environments for compiling, debugging and testing. Hardware designers usually communicate with layout drawings and circuit specifications, while mechanical engineers and designers utilise CAD (Computer-Aided Design) models and tools.

The multi-disciplinary nature of products brings new challenges to product development and management. As the number of design teams increases, the need for both informal and formal communication increases. Effective means of communication are needed to ensure the seamless co-operation of different teams, and to integrate different technologies toward a common target object. From the human resource management point of view, people that understand larger entities and several design domains have become critical.

At present, the technical and the methodological solutions of most engineering disciplines are already in practical use and available to every company. There are also various design tools for industrial and mechanical design, software and hardware engineering, product data management, sales and financial management. In practice, the problem is not to find a tool for a specific task, but rather to find tools that are compatible with each other and can in a conjoint effort support fluent execution of product development and business processes.

The solution for a successful product development is not any more based only on advanced tools and methods of different design areas – more often the key issue will be an effective control of the product development process, and an effective management and integration of knowledge via various tools and engineering approaches (Davenport and Prusak 1998).

2.4 From local to global operations

The shift towards global markets and organizations has opened new possibilities for companies. Along with the expanded market potential, companies have the opportunity to widen their operation and increase turnover and profit. Likewise through global operations, companies can balance their total market share during regional depressions and ease their resource and knowledge needs.

On the other hand, global and distributed business set entirely new demands for organizations and also for product development. Effective communication is needed between geographically dispersed development teams. Rahikkala (2000), for example, has studied how software development can be organized in distributed, dynamic teamwork environments called virtual software corporations (VSC). Time difference and distance are the first obstacles, but there are also linguistic and cultural challenges in daily work that must be overcome. From the product marketing and development point of view multi-national markets set special requirements for the internationalization and localization of the products. Internationalization[12] must be taken into account by preparing products so that they can handle multiple languages and cultural conventions without the need for re-design. Localization[13], on the other hand, requires a careful study of local user preferences. It is not only the local language that must be taken care of, but there can also be cultural and geographical likes/dislikes for specific designs and product features on different areas and countries that must be recognized in the marketing. In addition, the product must be configured to fit to the local regulations and laws.

[12] Definitions for internationalization, localization and translation terminology are available at LISA (http://www.lisa.org), which is a non-profit organization of the GILT (Globalization, Internationalization, Localization, and Translation) business community.

[13] The Localization Industry Standards Association (LISA, http://www.lisa.org) defines localization as follows: "The process of modifying products or services to account for differences in distinct markets."

3. Development of electronic products

Personal electronics represent a relatively new product category of electronic products. This is why the design practices, methods and tools applied in this area are usually adopted directly from more traditional products and design areas. Especially software and systems engineering approaches have a strong influence on the design practices applied in this area. In this chapter, an introduction to the typical practices in the development of electronic products is presented. The work is started by analyzing the basic need that is common to the business area regardless of the selected design approach or process model – namely the need for effective communication and co-operation. After that an introduction to the product development processes and tools coming from the mainstream design approaches of software and systems engineering is given. Additionally, a study of newer design approaches emphasizing the UCD is conducted. Also the computer-based tools supporting development tasks are discussed, and an analysis of the role of prototyping and simulation technologies in product development is presented.

3.1 Demand for communication and co-operation

Modern electronic products are outputs of multi-technology development processes incorporating several engineering disciplines. Whether the development work is performed internally within a company, mainly using its own resources or through a subcontractor network, it is obvious that successful organization and management of this work relies on effective communication, co-operation and co-ordination between the developers.

In practice, the problems in communication and co-ordination appear at two different levels, the human and technical levels.

Human level. Product development, including joint work with marketing, manufacturing, and management activities (Ulrich and Eppinger 1995, 2000), is a highly interactive process involving individuals with different objectives and capabilities. This human factor, together with the complexity of the work in general, as given in (Tuikka 1994, Carstensen et al. 1995), uncertainty, dynamism, mutually interdependent actors and many highly interconnected product parts, make it difficult for participants, even in a small-sized team to fully understand the objectives of the work and the target product concept.

From the company viewpoint, the lack of understanding caused by ineffective communication and co-operation will result poor product quality, high development and manufacturing costs and extended development time. To avoid or minimize these effects, several approaches have been presented to support effective communication and co-operation in product development, especially under the terms of the CSCW (Computer-Supported Co-operative Work) (Carstensen et al. 1995, Johansen 1988, Bannon and Schmidt 1991).

Technical level. The obstacles for effective product development of multi-technology products are often caused by difficulties related to the integration of diverse engineering disciplines. As most engineering domains have their own modelling conventions and tools, the communication and co-ordination of the work may be difficult between the project participants. For a company, misunderstandings during the development work can be costly, especially if they are only recognized in the late product integration stages.

Effective communication is a base element of successful product development. In practice, the goal to attain better communication is supported by three elements of a company, namely, processes, tools and organization (cf. Figure 1). Process describes the sequence of operations to fulfil a task, and as such it also describes a model for communication inside the process. Process approach answers the question “What are we communicating?”. Tools, on the other hand, are needed to help the communication, and they answer the question “How to communicate?”. Finally, the organization model defines the addresses for the communication and answers the question “With whom to communicate?”. Although product development is easily seen as an embodiment of technical communication, we must remember that in practice it also contains a significant amount of human level communication.

To overcome the barriers and misunderstandings between various design disciplines, several solutions have been introduced. In general, at least the most critical mistakes can be avoided by an organization by adopting a product development framework that supports quality management actions, such as suggested in, e.g., Quality Function Deployment (QFD) (Akao 1990), Total

Quality Management (TQM) (Bounds et al. 1994) or ISO 9001 (2000). At the engineering level, the possible solutions mostly rely on some intermediate presentation formats in which the domain-specific presentations can be linked more easily to an understandable concept. Visualization and prototyping techniques are good examples of this approach. Visualization of the target product, whether using 2-dimensional (2D) images or physical models, has been seen as a simple yet effective way of helping the integration of different development functions since the early days of technology. Furthermore, prototyping has represented an even more advanced approach, offering the possibility to approximate the product in one or more dimensions.

3.2 Product development process models

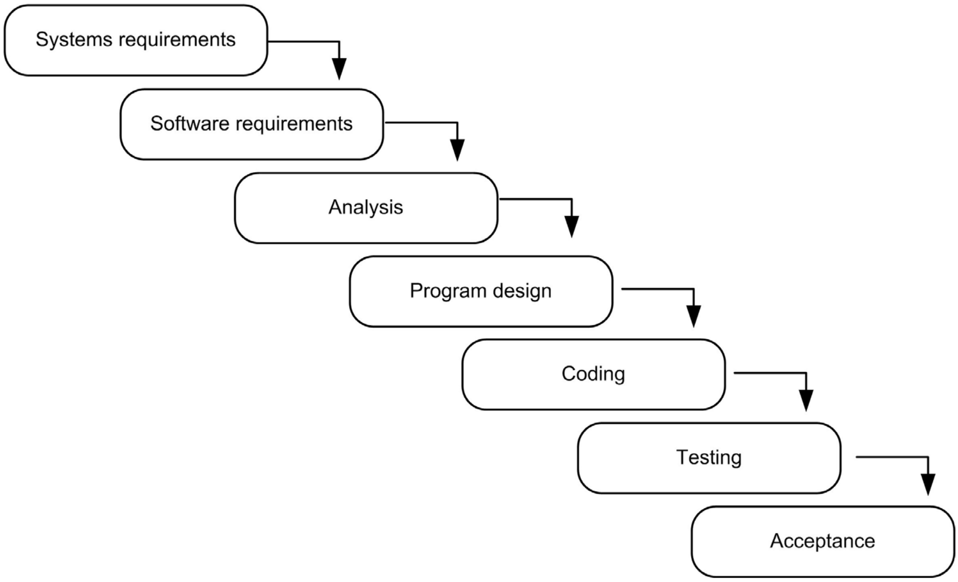
During the last twenty years, several process models have been presented for product development by researchers and practitioners. These models have originated from different backgrounds and for different purposes. As a coarse categorization process models can be divided into software engineering and systems engineering originated groups, which are described in more detail below. This categorisation is further extended with newer process models supporting UCD approaches.

3.2.1 Software engineering approach

Software engineers have especially been active in defining models for their work, and along with the growth of the software industry and applications, many of these have also been adopted in other engineering domains. The widely referred to classical waterfall process for software development (Royce 1970), on the other hand, was based on the sequential workflow introduced in more traditional industries in the 1960s. This model made the assumption that every development phase could be clearly separated and performed in a predefined order. No iteration was possible, which caused control problems in software projects. To overcome this limitation, several extensions, such as (Boehm

1986)[14], were appended to the model to allow for more flexibility to the task sequences. An evolvement of the waterfall model is illustrated in Figure 3, which shows how a basic waterfall model (Figure 3a) is still applied in the ISO/IEC 12207 international standard (Figure 3b) for software life cycle processes (1995).

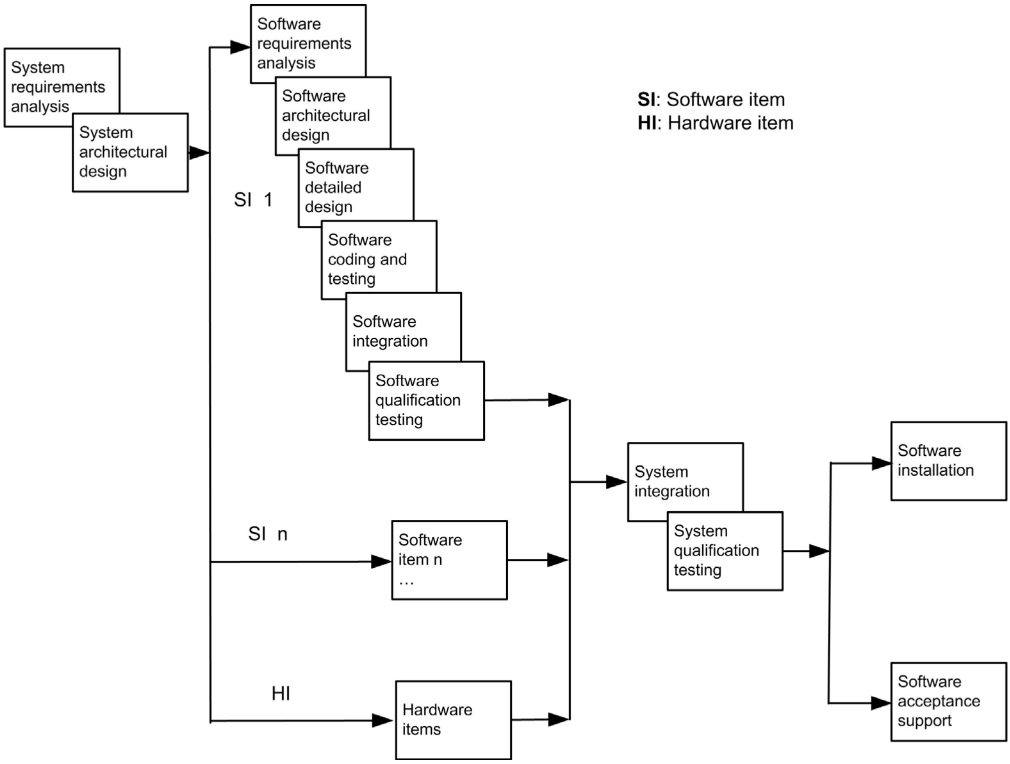
Rapidly evolving computer technology, together with growing quality concerns, new applications and new products, such as embedded computer systems, has resulted in new approaches in software process practices. It was recognized that the sequential approach was not the only way to develop software. Likewise, developers admitted that the process models were always, more or less, application and technology dependent, which led to the evolution of software process meta-models or frameworks.



a) Basic waterfall process model.

---

[14] Reprinted in 1988 in: Boehm, B. W. 1988. A Spiral Model of Software Development and Enhancement. IEEE Computer 21, Vol. 32, No. 5 (May), pp. 61–72.



b) Waterfall process model in ISO/IEC 12207 – Software Life Cycle Processes.

Figure 3. Classical waterfall software development process model.

Prototyping, and incremental and evolutionary development were seen as other promising software development approaches already in the 1980s. Software prototyping provided a way to build a prototype model or prototype components of the target software in order to obtain sufficient information for its implementation (Pressman 1987). Three different categories for software prototyping were defined by Agresti (1986):

- Behavioural prototyping supporting requirements analysis, for example, see (Vonk 1990, Dearnley and Mayhew 1983). Behavioural prototyping is also often referred to as rapid software prototyping or rapid prototyping.
- Structural prototyping for modelling and exploring the internal structure of the target system. For more information see, for example, (Gabriel 1989, Hekmatpour and Ince 1988).

- Evolutionary prototyping that enables the evolution of a prototype to a final product. More information on this can be found, for example, in (Lantz 1986, Connell and Shafer 1989).

Heterogeneous prototyping, combining previous categories, i.e. prototypes from different abstraction levels to an executable system model, has also been introduced, for example, by Gabriel (1989), Pulli[15] (1991) and Alonso et al. (1995).

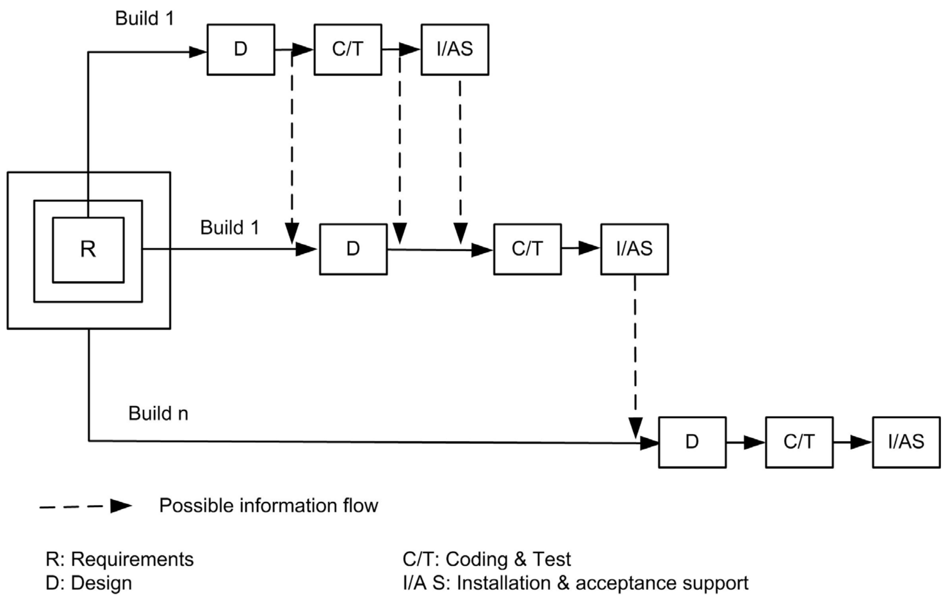


Figure 4. Incremental software development (source ISO/IEC 12207 – Software Life Cycle Processes).

Incremental and evolutionary models introduced yet additional approaches to software development. Incremental development means that a final software program is produced through the overlapping and iterative definition, design, implementation, and testing phases (Figure 4). It does not, however, modify requirements for each build, but the work is based on the same initial

---

[15] Pulli has given good general reference material for software prototyping in his doctoral thesis (Pulli 1991).

requirements. Evolutionary development, instead, also redefines or adjusts the requirements during the evolution of the builds (Figure 5). In general, the incremental model has been supported a lot by object-oriented design which provides many techniques for incremental software development, such as design abstraction and feature inheritance of the objects.

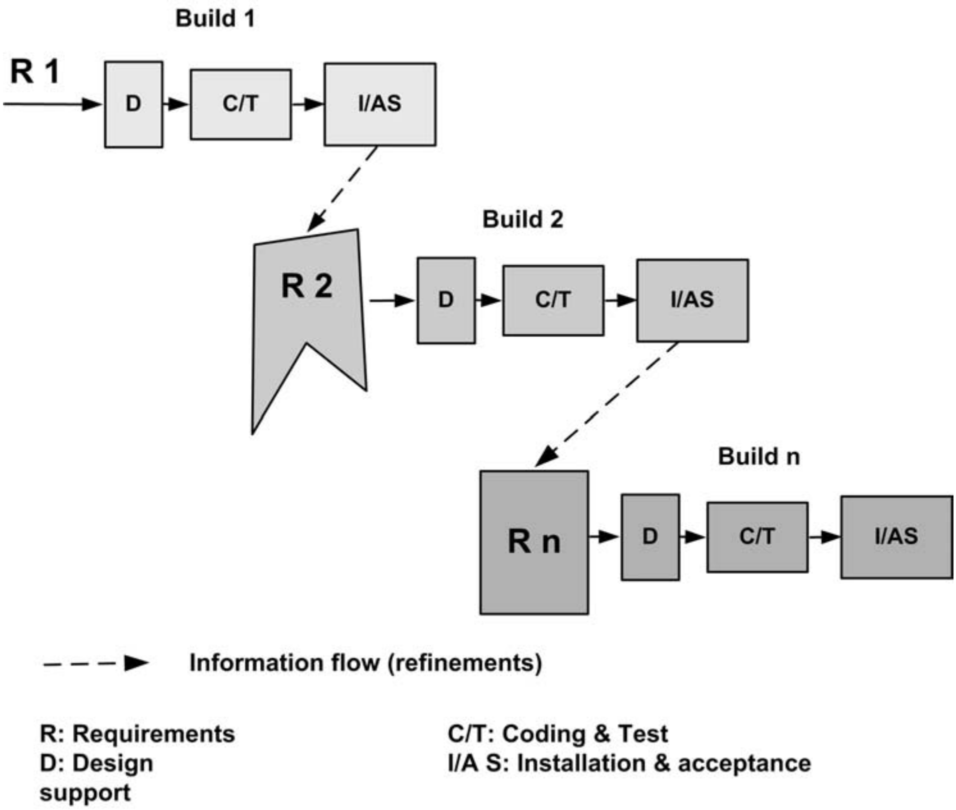


Figure 5. Evolutionary software development (source ISO/IEC 12207 – Software Life Cycle process).

Incremental and evolutionary development have possible links to prototyping, as prototyping can be applied inside the development (cf. evolutionary prototyping as the implementation of evolutionary development) or as a supporting approach (cf. rapid prototyping as a requirement analysis tool for incremental development). This shows that both models actually function as software process meta-models, i.e., they form a conceptual framework under which some selected software design approach or process model can be applied for specific tasks.

One well-known development framework was introduced by Boehm in 1986. His spiral model (Boehm 1986) provided project managers and developers a framework that helped them to justify the risks and strategy for the next development phase. Each phase could consist of the waterfall, prototyping or whatever development approach that was appropriate for the prevailing situation. In practice, the spiral model is an instance of evolutionary development, i.e., it conforms to the process presented in Figure 5, but in spiral form extended with risk management and the next phase planning tasks.

As given in Figures 3, 4 and 5, the waterfall, incremental and evolutionary process models are also a part of ISO/IEC 12207 international standard for software life cycle processes (1995). There are also many other international and national standards in software engineering that are linked to process related issues, e.g. from the quality (ISO 9000-3 1991), maintenance (IEEE 1219 1993), or configuration management (IEEE 828 1990) point of view. At the European level, the German standard for the v-model (BWB General Directive 250 1992) is worth noting, along with its “forerunner” the VDI Design Handbook 2221 for a systematic approach to the design of technical systems and products (1987). VDI 2221 is a guideline for general product design (cf. next chapter), and as such it is not very well-known among software engineers. However, the handbook also introduced a case example for software development and set forth an important initiative by acknowledging the role of software in systems and product design.

At the end of the 1990s software engineering processes also faced the challenges set by new businesses highlighting the role of fast software development cycles, and the importance of user involvement. The changes were launched mainly by the requirements introduced by high technology and software-focused businesses. It was noticed that software development methodologies introduced in the 1980s and 1990s were too disciplined to follow the dynamics and uncertainty of these business areas. New development projects were often focused on totally new technologies and solutions. The deadlines were strict, and the requirements were changing throughout the project. Often a project was needed to start even without knowing its final results. To meet these challenges new agile software development methodologies were introduced (Cockburn 2002). The main values of agile development are clarified by the Agile Alliance (2002) in the Agile Manifesto:

- individuals and interactions over processes and tools,
- working software over comprehensive documentation,
- customer collaboration over contract negotiation, and
- responding to change over following a plan.

In practice, agile methodologies rely on the flexibility and adaptivity of the development work. They emphasize working software as the main measure of progress, and aim to minimize the risk by developing the software in short iterative increments. When considering agile methodologies from the process model point of view it is worth noting that agile development, in fact, aims away from exact process model executions and guidance. This is why agile development prefers using the term ‘methodologies’ instead of ‘processes’. There are several methodologies introduced under agile development. The best known include Extreme Programming (XP) (Beck 1999), Scrum (Schwaber and Beedle 2001) and Feature-Driven Development (FDD) (Palmer and Felsing 2002).

Agile methodologies introduce important progress in software engineering also from the point of view of the development of electronic products. The methodologies recognize the change that occurred when software development was moving from large-scale to dynamic small-scale and user-focused projects and products. Personal electronics, indeed, represent a business area in which rapidly changing user preferences, fast progressing technology and global competition set high demands for the flexibility and adaptivity of the development methodologies. In this sense, this domain appears as one of the most potential beneficiaries of the agile approach.

3.2.2 Systems engineering approach

Systems engineering represents a traditional view of product development processes. Its roots date back to machine design at the end of the 19th century, and it has evolved along the developments in avionics, nuclear and space technology, and also in high technology electronics[16]. There are several definitions for systems engineering. Standards such as the IEEE 1220 Standard for Application and Management of the Systems Engineering Process (1998) and ANSI/EIA 632 Standard for Systems Engineering Processes (1999) have their own definitions, and, for example, US Federal Standard 1037C for Telecommunication Terms (FED-STD-1037C 1996) defines systems engineering[17] as follows:

- A process of defining the hardware and software architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.
- The preparation of an assembly of methods, procedures, or techniques united by regulated interaction to form an organized whole.

IEEE 1220 Standard for Application and Management of the Systems Engineering Process (1998) defines systems engineering as follows:

> *"Systems Engineering is an interdisciplinary collaborative approach to derive, evolve, and verify a life cycle balanced system solution that satisfies customer expectations and meets public acceptability."*

In this thesis, this definition is adopted with the exception that the term system can be equalled to 'product'. It should also be noted that in the context of the thesis the term 'systems engineering' includes many other product development approaches, which are often expressed in general terms, such as, product design, product development, engineering design, system design and system development. A common, distinctive factor of the previous terms compared to software engineering processes presented in the previous chapter is that they have no special focus in software development, but rather software is regarded only as one component or part of a product or system.

---

[16] For more information about this progress see, for example, Marcus, I. A., Segal, P. H. Technology in America: A brief history. Harcourt Brace Jonavich. Orlando, Florida, USA. 380 p.

[17] The standard uses systems engineering and systems design as synonyms.

From a process model perspective, systems engineering has a background in the classical sequential development process (cf. waterfall process for software development in Chapter 3.2.1). This sequential approach formed the first generation of product development process models, and was adopted, for example, by NASA in the 1960s, when the Phased Project Planning (PPP) process was defined for their internal use[18]. Stepwise development is still applicable with relative simple products and organizations. However, when the complexity of the design target and the number of involved design teams increases, the need for design iterations will become evident, and also where other approaches will be needed. Regardless of its limitations, the sequential process model functions as a good reference for a generic product development process model. It provides a way to learn the basic terms and methods of product development, and a template for further developing the process towards more sophisticated or customized models.

As in software engineering, the sequential process model has evolved into several new process approaches. The main reasons for this evolution have been the increasing complexity of the products, the tightening of competition resulting in a demand for shorter development times, and the globalization of both markets and organizations. The VDI Design Handbook 2221 (1987) was one of the early successors of the sequential process. It presented a systematic view to product development by dividing product design into subtasks in which sequential steps with possible iterations were applied. As examples of the approach design cases from mechanical engineering, process engineering, precision engineering and software development were given. As the VDI handbook 2221 was developed by German experts and supported by German authorities, it gained a significant role as a design guideline especially in European German-speaking countries.

Concurrent engineering (CE) has introduced another significant evolution of the sequential process model. CE basically refers to parallelization of product design tasks and phases in order to reduce development time. Because of this definition

---

[18] The sequential process model, in fact, also forms the basis for the second generation product development processes, and the CE approach represents the third generation processes. For more information about process generations, see for example (Cooper 1994, Cooper 1998).

CE has also been referred to as simultaneous engineering in many sources. From its initial approach solving problems of how to in parallel organize and carry out concept design, detailed design, testing and documentation actions, CE has been extended also to include business operations and phases outside the product design itself, i.e., how to design a product so that it also supports efficient manufacturing, logistics, etc., and how a company should be organized to support CE. One of the most referred definitions of CE describing the latter "CE in the large" approach is given by the Institute for Defence Analysis (Winner et al. 1988):

> *"Concurrent engineering is a systematic approach to the integrated, concurrent design of products and related processes, including manufacture and support. This approach is intended to cause developers, from the outset, to consider all elements of the product life cycle from conception through disposal, including quality, cost, schedule, and user requirements."*

CE has its origin in large-scale industries, such as military systems (Winner et al. 1988), ship building (Elvekrok 1997), automotive industry (Evans 1988, Thomson et al. 1987) aerospace and space industries (Burgess 1991, Ashley 1992), from where it has also expanded to electronics and computer design (Zimmermann 1997, Noore and Lawson 1992, Eppinger et al. 1994) and even to software development to some extent (Pulli and Elmstrøm 1993, Yeh 1992). Along with its success, CE has grown into a large sub-area of systems engineering, and has evolved into several new approaches such as Integrated Product Development (IPD), and Concurrent Enterprising and Design for Manufacturing.

When discussing product development processes, it is worth remembering that processes are always dependent on the given organization and business domain. This means that process models have to be customizable and adaptive to changing organization and business conditions. Some reasons for customization are mentioned, for example, in (Heikkinen 1997) and in the VDI Design Handbook 2221 on page 6 (1987). Ulrich and Eppinger (1995, 2000) also list versions of the generic process model for different product types, i.e. technology push products, platform products, process intensive products and customizable products. Electronic and telecommunication products can, basically, belong to any of the previous. However, when considering the area of personal electronics, emphasis should be placed on customizable products using design platforms. Platform design (Meyer and Lehnerd 1997) is widely adopted today not only for traditional consumer products, such as CD and video players, and television sets, but also for mobile phones and software products.

Platform design can be understood as a meta-process or process framework. On product level, it means that different product versions or variants are built with a minimum of changes from the same product platform. For example, a more expensive model of a DVD player has the same component structure and mechanisms as a cheaper one from the same trademark. But, in the expensive model some extra features are implemented either with a different software configuration or with an extra component. Platform design has a close link with mass customization (Pine II 1993, Tseng and Jiao 1998), which basically means the personalization of products with low cost. For example, mobile phone manufacturers realize that mass customization can be made effective by offering personalized product features and different product versions of same platform for various user groups.

On organizational or business level, platform design and mass customization often involve a lot of different product features and components being developed by R&D people. However, not all of these are selected for the final products, but the selection is performed by a product line or marketing responsible, who selects the appropriate features based on the company's technology and business roadmaps, market studies, usability tests, and the like. This kind of component, feature and concept pool approach is applicable especially to innovative companies with large R&D departments. Although a lot of research has been conducted on organizational and management issues of mass customization and product platform design, most of the studies have concentrated on the organization and co-operation of platform and product design teams. For an example, see Chapter 5 in (Meyer and Lehnerd 1997). There are, however, only few papers describing platform design from the R&D point of view, which could offer real solutions for product design teams and thus also affect the company's product strategy by demonstrating the possibilities of new technologies.

Systems engineering has also been evolving along the progress of products and businesses. Firstly, along the development of CE it was noticed that sequential development models hardly ever work as such. There are nearly always backward flows of information in system design, which gives rise to the need for design iterations and updates of requirements. Additionally, it was seen that the parallelization of the tasks and component development was needed in larger projects. Secondly, systems engineering practitioners have also recognized the increasing role of software in products. For example, Stevens et al. (1998) emphasized the role of software as a competitive factor in systems and defined two software dominated system types: software-intensive systems (SIS) and software-shaped systems (SSS). The former refers to large software systems that need to benefit systems engineering practice, and the latter denotes complex systems including a mixture of software, hardware and people (Stevens et al. 1998, p. 231). As examples of the latter SIS cameras, cars, planes and mobile phones were mentioned. Thirdly, systems engineering people have become aware of the importance of the user and user satisfaction in many highly competing business areas. For example, Stevens et al. (1998) recognize the role of UI and, more generally, that of the visual metaphor of the product in system requirements and design processes. A more concrete example of the integration of usability and the CE approach is given by Ketola (2002), who describes how usability and CE are integrated in mobile phone development at Nokia.

3.2.3 Process models for user-centred design

The role of good usability and user satisfaction started to receive increasing interest at the beginning of the 1990s. The issues of "how easy or effective is it to use a device?" or "how to make this product easier to use?" had already been studied before, after World War II, from the viewpoint of ergonomics. The focus then was mainly on products including mechanical controls. Later, along the development of computers and information systems, the focus of usability has been moving towards software systems and software-based products. In the 1980s and at the beginning of the 1990s these products were obviously not so easy to use. Several noteworthy books and articles (Norman and Draper 1986, Norman 1988, Shackel 1991, Hix and Hartson 1993, Nielsen 1993) defining the needs for new design practices recognizing user needs and usability were written at that time.

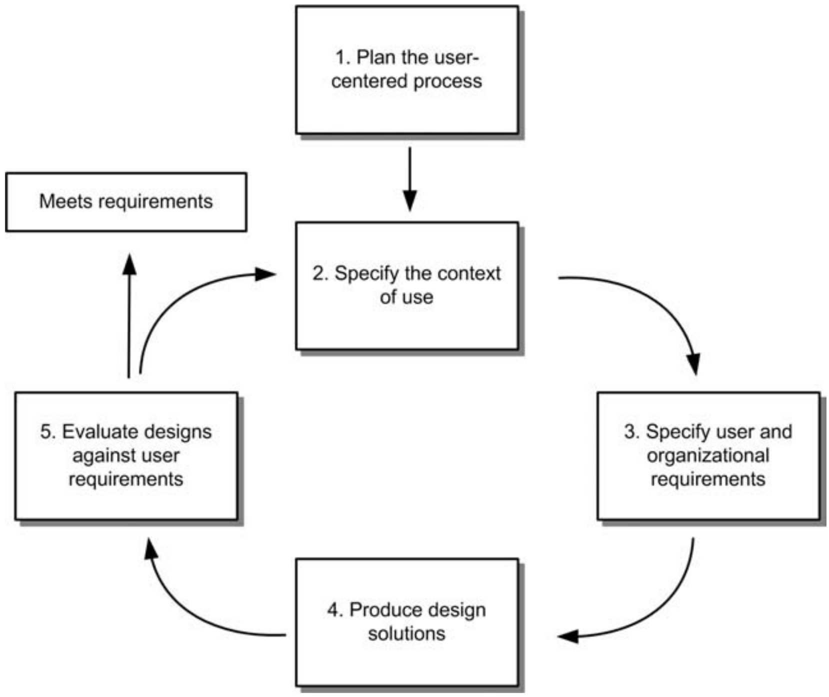


Figure 6. User-centred design process as defined by ISO 13407.

As a solution for achieving better usability in products, a new design process model emphasizing the role of the user and user needs was proposed. User-centred design (UCD)[19] was introduced as an international standard ISO 13407 "Human centred design process for interactive systems" (ISO 13407 1999) in 1999. The standard defined UCD as "an approach to interactive system development that focuses specifically on making systems usable. It is a multi-disciplinary activity". ISO 13407 lists the following principles for UCD:

- active involvement of users and clear understanding of the user and task requirements
- appropriate allocation of functions between users and technology
- iteration of design solutions
- multi-disciplinary design.

[19] Also known as human-centered design.

Implementing "iteration in design solutions" is based on the UCD process, which is illustrated in Figure 6. The process specifies the following activities:

- planning of human-centred design process
- specification of user and organizational requirements
- understanding and specification of the context of use
- production of design solutions
- evaluation of designs against the requirements.

Even though there exist many other UCD process models, most of the models share the common principles that are presented in ISO 13407 UCD. For example, many large companies and organizations have their own UCD process models[20]. One modification of the given UCD is presented by Jokela (2001), who modified the model according to the experience attained from five industrial cases of UCD assessment. Another example of a modified UCD process is the Delta method developed in co-operation by Ericsson Infocom Consultants AB and Linköping University (WM-data AB and Ericsson Radio Systems AB 2005). The Delta method proposes an approach in which the actual UCD activities are supplemented with the existing product development process (Figure 7). These approaches mainly contribute to the requirements analysis and the initial design phases. Additionally, usability testing can also be performed as a part of the final system testing phase.

The UCD process has been applied and researched especially in the development of software products and systems. The reason for this is that UCD has a strong background in this area. In the history of UCD, especially the interaction and UIs of information systems and software products have been in the focus. When comparing the software and UCD process models, it is also easy to see that UCD fits well to the software design process. UCD can be implemented beside the software process, so that it outputs material, such as usability requirements, UI specification, UI design and UI implementation, to the main process. In software design the success of UCD is easily seen as a success of UI and the interaction style of the developed software.

---

[20] For example, IBM introduces its UCD practices in web http://www-306.ibm.com/ibm/easy/eou_ext.nsf/publish/558. Similarly, NASA has information on its UCD activities in http://www.grc.nasa.gov/WWW/usability/index.html.

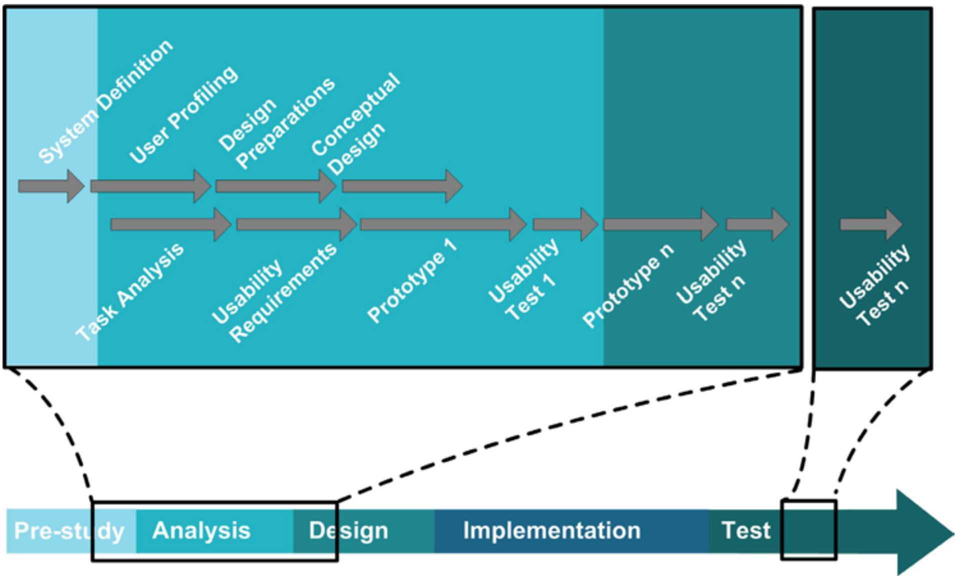


Figure 7. The Delta method from Ericsson Infocom Consultants AB and Linköping University.

When UCD is applied in the development of electronic products, its implementation becomes more demanding. The usability of an electronic product does not only depend on software, but it also integrates the results from mechanical and industrial design. Similarly, the development process applied is not any longer a software development process, but is going in the direction of systems design practices. Practical challenges arise with the questions: How to implement a UCD process? Should we run a product development process inside the UCD process, or should we shift the UCD process towards our existing product development process? Implementing UCD, for example, in a product project executing a CE process and having a tight timeframe is not an easy task.

Considering the practical implementation of UCD, it seems that customizing the current product development process in order to take into account UCD principles and activities is the best way to proceed. This way product development can be adjusted in order to recognize the importance of the early requirement phase specifying the context of use, as well as the user and organizational requirements. The requirements are further validated with active concepting. During the design and implementation, the process is further supported with prototypes, usability tests and recommendations. The approach of integrating UCD to an existing product development process is adopted by many of the UCD models and their industrial implementations. The Delta method explained above is one example of this. Another interesting example is given by Ketola (2002), who studied the challenge of integrating usability and UCD to a CE process in mobile phone development. The main results of his thesis was “to amend and supplement Concurrent Engineering with usability engineering activities, reported in the form of a Usability Engineering Guideline”.

UCD may not be considered solely as the guidelines behind development, but it should also be incorporated into practical work and everyday tasks. Based on the UCD literature, in general, and in the findings of Ketola (2002), it can be summarized that in order to successfully implement UCD and usability engineering in the multi-disciplinary development process of electronic products the following prerequisites should be met:

- UCD must be supported by company management.
- Usability engineers must have the authority to work in the product development.
- Product development teams must be aware of the implemented UCD and its goals within their work and in the company’s business overall.
- A usability plan for each development project must be created and integrated into the project plan.
- Usability engineering must cover all of the product parts and design areas that contribute to the usability of the product.
- UCD should concentrate on the early development process phases of product concept design and requirement analysis.
- It is best if the early UCD activities of the context study, as well as user and organizational requirement studies can be conducted outside and before the actual product development.
- Usability tests benefiting different concept and product visualizations and prototypes are used in planned process phases and milestones.

As a final remark on UCD, it is important to pay attention the role of prototypes. Prototypes are mentioned by all of the UCD approaches and guidelines as practical tools for acquiring user feedback in usability tests. Depending on the target product under development and the phase in which the prototype is applied, different prototyping approaches can be used. The literature and research list many possibilities for prototyping, including high-fidelity and low-fidelity prototypes, hardware mock-ups and mechanical simulations, software prototypes, paper prototypes, feature prototypes, virtual prototypes, rapid prototypes and a prototype series. These approaches are studied further in Chapter 3.4.3.

3.3 Computer-aided tool support

Nowadays, product development is supported by an enormous number of computer-aided design tools. One way to categorize these tools is to divide them into actual design tools, design analysis tools, design support tools and other supporting tools. An exact division of the tools, however, is not possible because some of them can cover more that one of the aforementioned groups.

Design tools are productive tools that contribute real components or parts to the product under development. These tools can be further categorized as follows:

- *Computer-Aided Engineering (CAE)* is a general description for tools that use computers in the design, analysis and manufacturing of products.
- *CAD (Computer-Aided Design) tools* basically include all of the design tools that use computers. With this wide definition most of the below mentioned tools also belong under the term CAD. However, CAD also has a more restricted denotation originating from mechanical engineering, under which it means "Computer-Aided Mechanical Design". In this thesis, the latter definition of CAD is adopted, i.e., CAD refers to computer-aided tools for mechanical design, including structure, dimension and material design. Today, most CAD tools are capable of 3-dimensional (3D) design. Among the most common CAD tools are AutoCAD (http://www.autodesk.com) and ProEngineering (http://www.ptc.com).
- *Computer-Aided Software Engineering (CASE)* refers to the tools that are used for software and software component design. These tools include code editors, linkers, compilers and integrated development environments (IDE).

CASE tools also include design tools supporting different software design notations or formalisms, code generators and graphical user interface (GUI) design tools. It should be noted that in certain contexts CASE may also refer to Computer-Aided Systems Engineering. Among the best-known CASE tools are, for example, StateMate (http://www.ilogix.com) and Prosa (http://www.prosa.fi).

- *Electronic Design Automation (EDA) tools* are used for different electronic design tasks, such as the designing of ICs (Integrated Circuit), PCBs (Printed Circuit Board), along with packaging, wiring and embedded software design. This tool group is sometimes also referred to by the initials CAEE (Computer-Aided Electronics Engineering). As tool examples, Cadence Design Environment (http://www.cadence.com) and Mentor Graphics' design tools (http://www.mentorgraphics.com) can be mentioned.
- *Industrial design tools* include sketching and visualization tools that are used mainly in the concept design phase. These tools are often included in CAD tools or have interfaces to them, so that their outcome can be directly made use of in detailed mechanical design. Some tools can also use the web for collaborative design sessions and for virtual reality (VR) applications for more realistic visualizations. The term CAID (Computer-Aided Industrial Design) has also been introduced for this tools group. As product examples, in addition to the modules offered by CAD systems, Alias Wavefront Studio Tools (http://www.aliaswavefront.com) and Opticore Opus Realizer (http://www.opticore.com) can be mentioned.

Design analysis tools provide analysis functions for verifying design results. Even though many design tools include analysis functions already in their basic packages or as separate extensions, there also exists a separate tool for this purpose. Examples of this category are:

- *Analysis, simulation and prototyping tools for mechanical design* performing strength and stress analysis, simulation of moving parts and calculation of weight. These tools are often integrated to or offered as extensions to popular CAD systems. Examples of this category include Moldflow Plastics Advisers (http://www.moldflow.com) and Dynamic Designer from Design Simulation Technologies, Inc. (http://www.design-simulation.com).

- *Analysis, simulation and prototyping tools for electronics design* providing logic and timing analysis and simulations. Board layouts and wirings can also be simulated in order to verify electromagnetic properties such as crosstalk, radiation and delays. Thermal analysis tools are used alongside software testing tools to test the embedded software of electronic components. As in the CAD area, these tools or features are often offered as a part of common EDA tools. Examples of these include ModelSim tools from Mentor Graphics (http://www.mentor.com) and MultiSim from Electronics Workbench (http://www.electronicsworkbench.com).
- *Analysis, testing and prototyping tools for software engineering* that verify the proper operation of implemented software. The tools support different testing methods such as black-box and white-box testing, integration testing, regression testing, load testing and security testing. Products like TestComplete from AutomatedQA Corporation (http://www.automatedqa.com) and Rational Software tools from IMB (http://www-306.ibm.com/software/rational/sw-bycategory/subcategory/SW730.html) represent this tool category.

Design support tools help in managing and controlling the design work and design data, i.e. the output of design tools. Typical examples of these tools are:

- *Product Data Management (PDM) systems* are basically meant for managing and controlling product components, structures and their versions. Originally, they had a focus in finalized components, not in design data management, which they also cover today. PDM has a strong background in mechanical engineering, and does not usually focus on software components of the product. Examples of PDM systems are Windchill (http://www.ptc.com/products/windchill/index. htm) and Aton ODM from Modultek Oy (http://www.modultek.com).
- *Product Life Cycle Management (PLM) systems* extend the coverage of PDM systems towards organization and product life cycle dependent data. Basically, as PDM manages the product data, PLM manages how to use this data not only in engineering tasks but also in other business tasks. Typical functions offered by PLM systems include workflow, process and project management, communication support, data presentation services and information authoring tools. However, in practice, it is difficult to make any

exact difference between the features of PDM and PLM systems. PDM and PLM systems are usually offered by same system vendors. Product examples are Agile™ PLM from Agile Software (http://www.agile.com) and ProductCenter™ from SofTech Inc. (http://www.softech.com).

- *Software Configuration Management (SCM) tools* manage and control software configurations and components. They can be regarded as PDM systems for software engineering. SCM tools usually focus on design time control of the software components. Examples of SCM tools are Performance SCM system (http://www.perforce.com) and PVCS Dimensions (http://www.merant.com/products/pvcs/dimensions/index.asp).

Other supporting tools include various computer-based tools that can be of benefit in managing, controlling, communicating, documenting and securing within the development process. The following are some examples of these:

- *Office tools* are used for everyday documentation and communications. They include word processors, spreadsheets and presentation tools. An example of this group is Microsoft Office (http://www.microsoft.com).
- *Project and process management tools* help in project planning, reporting and follow-up. Typical tools include features for timetable planning and follow-up, and for resource management and allocation. Such tools are, for example, Microsoft Project (http://www.microsoft.com) and TurboProject by OfficeWork Software, LLC (http://www.officeworksoftware.com). Besides for project management, there are also tools for process management and enactment. Some of these focus on company level actions, such as those modules provided by PDM/PLM systems, and some in more specific areas, such as software engineering. For examples, see PMOLink's Project Management Process tool (http://www.pmolink.com) and ASG-Visual Process from Allen Systems Group, Inc. (http://www.asg.com).
- *Tools for groupware and CSCW* provide support for communication and collaborative design through computer networks and the Internet. At the minimum level, these tools include basic Internet and mobile communication tools, such as e-mail, newsgroups and web browsers. Examples of these are the Eudora e-mail software from Qualcomm (http://www.eudora.com) and Firefox web browser from Mozilla Corporation (http://www.mozilla.com/firefox/). More advanced systems can offer even

synchronous communication for a group of people to a shared workspace where people can comment and even change the properties of the target object. These kinds of systems can take advantage of Internet meeting tools, web cameras and some basic collaboration platforms, such as Lotus Sametime from IBM (http://www-306.ibm.com/software/lotus/) or BSCW from Fraunhofer FIT (http://bscw.fit.fraunhofer.de).

- *Drawing and image editing tools* help in providing illustrations, such as organization charts, workflow graphs, network diagrams and photos, for documents. Examples of these tools are Visio from Microsoft (http://office.microsoft.com) and Paint Shop Pro from Corel Corporation (http://www.corel.com).
- *Utility tools* are needed in various tasks. They can, for example, secure files and folders, protect a PC from viruses, compress files, take care of calendars, reserve meeting rooms, synchronize information between different devices, et cetera. For some examples, see F-secure tools from F-Secure Corporation (http://www.f-secure.com/products/), WinZip from WinZip Computing LLC (http://www.winzip.com) and PC Suites for Nokia mobile phones (http://www.nokia.com/phones/index.html).

3.4 Prototyping in product development

Prototypes and prototyping have important roles in the development of modern technology products. Despite this fact, the definition of prototyping is not definite, but there are several definitions for the term depending on the engineering or expertise area and even on different schools inside these. Because of its multi-disciplinary nature, prototyping and its supporting tools were not listed as a separate group in the previous tool categorization, but more as a horizontal supporting activity going through the different design areas. For example, CASE tools can include support for software prototyping, CAD tools support rapid prototyping and EDA tools also cover the virtual prototyping of ICs.

The Webster's Dictionary defines prototyping as:

> *"An original or model on which something is patterned and/or a full-scale and usually functional form of a new type or design of a construction (e.g. an airplane)."*

This definition also applies to product development. However, as Ulrich and Eppinger (1995, 2000) put it, prototype in product development has a wider meaning as it does not actually only refer to a noun prototype, but also to a verb and adjective; thus one may "prototype a design" and "make prototype components and systems". In general, a prototype in product development can be understood as an experimental and often incomplete model of the target object or system that approximate the target in one or more dimensions. Consequently, prototyping can be defined as a process of developing and using the prototype for different purposes. Basically, a prototype can be a physical model built manually, while presently it is usually built using computer-based tools. Or, it can be a digital model that only runs in the computer's memory. A prototype may also be a software module or programme, as it is understood in software engineering (Vonk 1990, Gabriel 1989). Today, heterogeneous prototypes combining the power of computer-based visualizations and simulations can be very efficient as several features of a product can be examined simultaneously.

Prototyping has close links to computer visualization and simulation, which can be regarded as background technologies for it. Similarly, virtual reality (VR) technologies have contributed considerably to many prototyping approaches. To shed light on the relationships of previous technologies and the possibilities of prototyping technologies in virtual design, separate chapters have been allocated for these issues.

3.4.1 Visualization and simulation

In general, visualization means the formation of visual images, the act or process of interpreting in visual terms or of putting something into visual form. Within this definition, different categories of visualization can be separated. Engineering or design related visualization originate from CAD or industrial design, in which the need is to represent a target object with an adequate accuracy of measurements or appearance in either 2 or 3 dimensions. Data visualization refers to the representation of abstract business or scientific data as images that can help in understanding the meaning of the data.

Simulation, like visualization, is a broad concept. Depending on the application area and the focus of the work, the expression has different meanings. Currently, the term simulation often refers to computer simulation in which a real phenomenon is imitated with a set of mathematical formulae and computer programmes. Typical examples of this are simulations of weather conditions, physics phenomena, integrated circuits and other hardware components. In software and system simulation, the objective is to validate and verify the designed software code or system components by replacing input data and possible missing components by their simulations. In this approach, simulation approaches have a lot to do with issues related to testing.

3.4.2 Virtual reality and interactive product models

The basic concepts of virtual reality (VR) and virtual worlds were introduced already in the 1960s within visually coupled systems (Comeau and Bryan 1961, Sutherland 1965). In those early days, VR was mainly regarded as implementations for special equipment and computer systems. During the early 1990s, the increasing power of personal computers (PCs) and workstations brought it closer to the general public. In 1992, Silicon Graphics Inc. (SGI) introduced the OpenGL 2D/3D graphics API (Application Programming Interface) (Segal and Akeley 1992). This can be regarded as the starting point for the development of virtual world authoring tools for Silicon Graphics workstations. On the PC side, Superscape (formerly known as Incentive) in the UK was started even earlier – the Freescape Control Language, which was originally designed to control virtual worlds and computer games, was ported to PCs in 1987. Based on this development, virtual worlds nowadays are more often experienced as desktop VR in which the virtual world is restricted to the area between the computer screen and the user.

Virtual worlds were originally understood as artificial environments generated for a user by a computer and VR technologies. The concept had many resemblances to computer games, in which the player was immersed in the game environment. In both approaches the goal was to represent an operating environment with an adequate degree of realism and accuracy. The development of Internet technologies also offered a networking aspect. Today it is thus possible to have networked virtual worlds and computer games where several people can participate in the same session and interact with the environment and each other.

The use of VR technology today in product development is mainly concentrating on product visualization and demonstration, and collaborative work support. The former approach means that virtual product models are shown in functional, interactive, photorealistic and 3D form, usually on a web page. The latter approach refers to a virtual space that can be used to communicate between virtual presentations of the participants, e.g. Avatars (Rheingold 1993, Benford and Fahlen 1993, Luciano et al. 2001). Additionally, product models and other artefacts can be placed in a common virtual environment where several people can interact with them (Tuikka 2002). Technically, collaboration can happen in a VR environment where participants are actually present or in a web-based networked session. As in groupware, the communication in a collaborative virtual environment can be asynchronous or synchronous (Maxfield et al. 1995).

VR and virtual models introduced two important dimensions for prototyping: functionality and interactivity. Compared to visualizations, simulations and typical 2D and 3D CAD models applied in design, the virtual models are able to contain even complex functionalities that are controlled by arbitrary user interactions. For example, 3D CAD tools can provide some basic mechanical functions and animations, such as opening a door or sliding a cover, but they do not usually provide means for building more advanced logical functions, such as "turn on the yellow light if the red light is on and a button is pressed". Virtual models can even include complicated logic, because, for example, for VRML (Virtual Reality Modelling Language) models functionality can be defined through VRML or Java scripting language, or through Java classes (Campione and Walrath 1998) or VRML External Authoring Interface (EAI) (ISO/IEC 14772-2 2004).

3.4.3 Classifications of prototypes

In order to evaluate the characteristics and the overall maturity of a prototype, different classifications have been introduced in which prototypes are categorized according to such dimensions as its scope, representation and executability. One basic approach is to classify prototypes according to their representation, i.e., if it is a physical prototype or a mathematical simulation of characteristics. For example, Ulrich and Eppinger (1995, 2000) use this dimension as one of the axes in their prototyping classification scheme. Another axis is formed by completeness, i.e., if a prototype focuses on one or several product characteristics; in other words, how comprehensive a model it is of a target object. Stevens et al. (1998), on the other hand, suggest vertical prototypes focusing on a small design area in detail, and horizontal prototypes providing a broad view of the system functionalities. Software engineering, again, introduces even more comprehensive classifications of prototypes. Throwaway, quick-and-dirty and evolutionary prototypes are some examples of these (Boehm 1986, Smith 1991, Wood and Kang 1992). In general, the separation of prototypes into low-fidelity and high-fidelity prototypes seems to be adequate for the categorization various applications. A low-fidelity prototype stands for a low-cost, limited in functions, limited in interaction and fast to build prototype. High-fidelity prototypes are finished, highly detailed prototypes that can provide a near realistic user experience and are typically more laborious to create compared to low-fidelity prototypes.

3.4.4 Prototyping approaches

Several prototyping approaches have been introduced for different applications and technology areas in product development. In mechanical engineering, prototyping is usually considered equivalent to rapid prototyping producing complex physical prototypes directly from 3D CAD models. However, in software engineering, prototyping is often tied to a software development process in which prototyping programmes are used in the early definition of customer requirements and in the continuous refinement of the development output, as presented in (Smith 1991, Boehm 1986, Pulli and Elmstrøm 1993, Bischofberger and Pomberger 1992). These and other more general prototyping terms and their application areas are described in Table 2. The table attempts to clarify these rather new, and often, confusing prototyping terminologies by presenting their general meanings, goals, and the synonyms used if available.

Virtual prototyping is a prototyping approach that can be applied in different design areas. Therefore, when referring to the approach it is important to mention which areas it is applied to. One of the basic articles defining the virtual prototyping approach for the research world was given by Haug et al. in 1993 (Haug et al. 1993). Even though Haug worked in the area of mechanical design and dynamic simulation of mechanical systems, he defined the virtual prototype and virtual prototyping on a general level, which made his definitions acceptable also for other design areas. Haug (Haug et al. 1993) defined the virtual prototype as:

> *"A computer-based simulation of a prototype system or subsystem with a degree of functional realism that is comparable to that of a physical prototype",*

and virtual prototyping as:

> *"The process of using a virtual prototype, in lieu of a physical prototype, for the test and evaluation of specific characteristics of a candidate design."*

Since the 1990s virtual prototyping has been used mainly in mechanical engineering and manufacturing, where the applications have usually been derived from visualisation and modelling tasks in CAD and CAM. The most successful applications have been in the field of the automotive (Potter 1996), military (Schaaf and Thompson 1997) and aerospace industries (Bennet 1997, Stytz et al. 2001), in which virtual prototyping has been applied in, e.g., mechanics and dynamics design, concept and design visualization and testing, UI design, assembly line modelling, and product assembly training.

Another strong area for virtual prototyping is electronics design, in which virtual prototyping is applied both in the design of individual electronic components, such as ASICs (Application-Specific Integrated Circuits) and FPGAs (Field Programmable Gate Arrays), and in the software-hardware co-design of electronic systems and components. The first approach focusing on individual components is often referred to as silicon virtual prototyping or silicon prototyping (Dai et al. 2003). In this approach, different tools are used to provide early estimations and information on the final physical implementation of different design alternatives. Characteristics such as timing, area and power of design choices can be estimated in advance to optimize the final design. In software-hardware co-design, the purpose of virtual prototyping is to use virtual prototypes of hardware components for testing and verifying the functions and performance of an embedded system, DSP (Digital Signal Processing), ASIC or SoC (System-on-chip) solutions together with software functionalities in early design phases. In many cases, the virtual prototypes of hardware components are presented as VHDL (Very High speed integrated hardware Description Language) models (Rundquist 1995, Valderrama et al. 1997). A good summary and status report on virtual and other prototyping techniques and tools used in electronic design automation is given in the European Medea+ research project (MEDEA 2003).

In addition to its main application areas in mechanical engineering, manufacturing and in electronics design, virtual prototyping has been applied in many other design areas and tasks. Because of the virtuality aspect of the approach, it offers interesting possibilities to model and test entirely new design and product ideas by using VR. Additionally, by combining virtual worlds and environments to virtual prototyping it is possible to implement collaborative design and prototyping environments. Virtual prototyping has been used in modelling and testing the UIs and usability of the products, e.g., in the fields of robotics (Fridenfalk et al. 1999) and haptic UIs (Buttolo et al. 2002, Dionisio et al. 1997). However, besides the work presented in this thesis and in the research projects behind it, there are only a few research reports available on applying virtual prototyping and prototypes as a multi-disciplinary and heterogeneous prototyping approach for software-focused consumer electronic products. One research project focusing on integrating physical and software prototyping in mobile phone design is presented by Loh et al. (1999). The authors introduce a PhonePro virtual prototyping platform, which enables a modelling of GUI, physical user interface (PUI) and dynamic behaviour of mobile phones with appropriate editors (Loh et al. 1999).

UCD has given rise to an increasing need for different prototypes and prototyping approaches. Prototypes are needed in usability tests guiding the development work under UCD principles. A lot of literature is available on prototype approaches that are used in UCD. Kiljander (1997, 2004), for example, presents a classification of mobile phone user interface prototyping methods (Figure 8). He argues that there is no single optimal prototyping method to be applied in mobile phone user interface development, but different methods need to be applied in different phases of the overall process. The use of low-fidelity prototypes has strong support especially in the early activities of UCD. They are fast to create and change, and as such they offer an effective way to explore the design alternatives.

Table 2. Prototyping approaches.

Prototyping Approach	Application Areas	Description	Goals	Supports User Interactions	Supported Functionality	2D Support	3D Support	Synonyms	Notes
Feature prototyping	User interface and software design	Simulates product behaviour with functional product models	To model, test and validate the functionality of the UI software and outlook.	Yes	Logical, i.e., UI behaviour	Usually			Quite new term introduced by Altia (http://www.altia.com).
Paper prototyping	User interface design, usability testing	The utilization of paper, not software tools, to simulate the use of software by target users.	To test quickly, cost-effectively and interactively new UI design.	Yes	Logical	Yes	No		Suitable for testing of relatively simple behaviours. Cannot simulate system response times.
Rapid prototyping	Mechanical eng.	Physical models and prototype parts are built directly from 3D CAD data.	To validate the design with low-cost physical mock-ups before production.	Physical touch	None	No	Yes		Typical technology: strereolithography, 3D printing.
Software prototyping	Software eng.	Enactable SW prototypes and different prototyping tools are used for validation and development of a SW system.	To validate the system requirements and/or to deliver a working system (evolutionary prototyping) based on prototypes.	Possible, but usually limited	Logical	For screens, user interfaces	None	Software rapid prototyping	
Virtual prototyping	Mechanical eng., CAD	Visualization of the target product and its environment in 3D. Can include mechanical, heat dissipation or mechanical dynamics simulations.	Usually aims to analyse the physical characteristics of a design.	Possible	Mechanical	Possible	Yes	Digital prototyping	
	Hardware design (ASIC, FPGA, PCB)	Hardware boards and components are designed by using SW and HW (usually programmable) prototype components.	To ensure the design with more flexible and cost-effective prototypes.	Not typical	Logical	Yes	No	Rapid prototyping	
	Multi-engineering product development	A product or a product concept, its behaviour and usage are simulated as realistically as possible by using simulations and prototypes.	To provide a realistic simulation of a target object (i.e., physical and logical functionality and appearance) and its possible usage situation.	Yes	Mechanical, logical, functional	If needed	If needed		
Virtual reality prototyping	Multi-engineering product development	Extends virtual prototyping so that a product or a product concept, its behaviour and usage can be simulated as realistically as possible by using different simulation models and VR techniques.	To provide a realistic simulation of a target object (i.e., physical and logical functionality and appearance) and its possible usage situation.	Yes	Mechanical (advanced), logical, functional	If needed	Default		

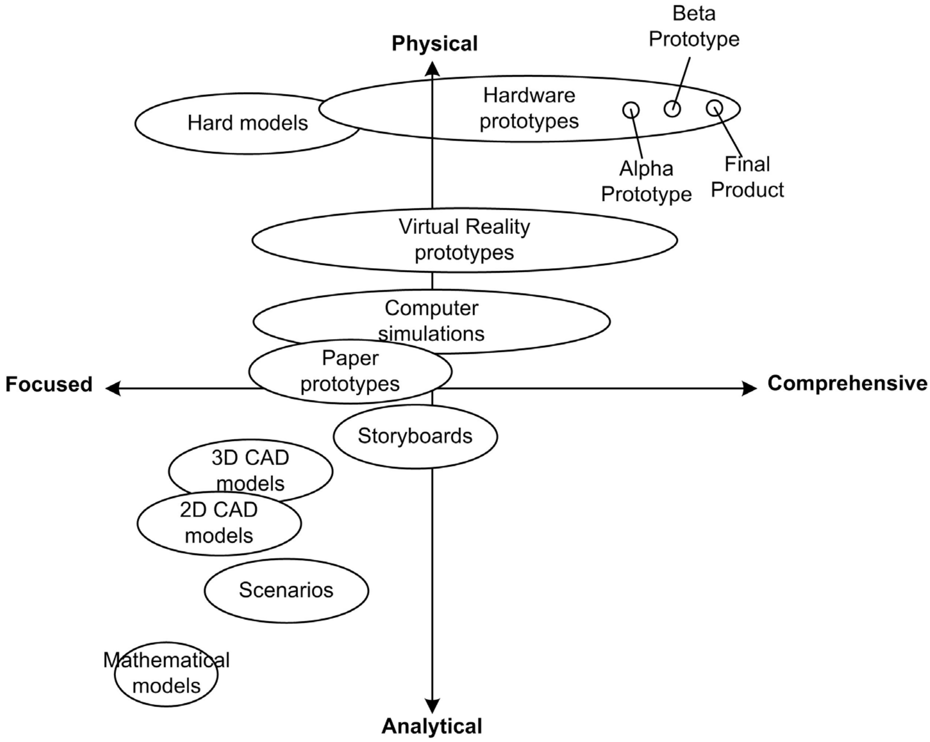


Figure 8. Classification of mobile phone user interface prototyping methods (Kiljander 1997).

One specific type of low-fidelity prototypes that are popular in usability engineering are paper prototypes. They are drawings illustrating the screenshots of the tested device or software. Drawings are presented to a test user according to the device or software logic by a test moderator representing the "computer". The user manipulates the papers by clicking buttons, typing text, et cetera, according to his or her task. The use and benefits of paper prototypes have been reported, for example, by Rettig (1994), Jokela and Pirkola (1999) and Ketola (2002). Ketola lists the main benefits of paper prototypes in mobile phone development as follows (p. 109, Ketola 2002):

- Paper prototypes enable reliable user testing of the interaction in the early development phases.
- The form and size factors of mobile phones are optimal for paper prototypes.

- Low-fidelity (paper, cardboard) mock-ups are not good for testing industrial design if they are produced manually.
- User testing with paper prototypes is efficient in producing new design ideas.

As a drawback for paper prototyping Ketola (2002) mentions that they cannot simulate dynamic communication very well. For example, a voice call interaction and other audio properties cannot be easily integrated into the prototyping session.

4. Requirements for enhanced product development

In this chapter the requirements for an improved product development process and its supporting tools are presented. The requirements are analysed based on the problems recognized in the case companies. After problem analysis, the general objectives for effective product development are given, i.e., the role of product development is tied to the overall company functions and trends influencing the electronic industry (see Chapter 2). In system or software engineering, these objectives could also be understood as requirements for the target system. Then, more detailed requirements for process realization are given (cf. specifications in system or software engineering) and finally, the requirements are set for enhanced design tools and prototyping support.

4.1 Problem areas

During the co-operation with the case companies, several problems or problem areas were recognized. Many of the them were tied to more fundamental problems, which occurred when development practices coming from software and systems engineering were applied in personal electronics design and business. Other problems were further categorized into limitations in tools and prototyping support.

4.1.1 Limitations of the process models

In general, the introduced process models serve well in their specific application or technology areas. For example, there have been successful, large software projects in avionics and the space industry that have been conducted using software engineering process models, but also as a subpart of systems engineering process of the entire system. However, problems with common process models have been encountered, especially in the following situations:

1. Process model is applied to new technology or business area, or a specific process model is applied to an incorrect business or technology area.

2. A certain process model is applied at an unsuitable company maturity level (entrepreneurial, growth, decline, renewal, established, et cetera) or in an unsuitable organization structure (e.g. SME organization, large national company, multinational company, or virtual organization).
3. General business or technology changes so fast that the organization and processes cannot adapt to them.

Examples of these situations are:

Example 1: A SME (PrE in Table 1) producing mechanical constructions was gradually involved in software design as the number of software controlled parts increased in their products. In the beginning, the design of relatively simple software modules did not cause any problems for the company. However, later, as the size of the software increased, the software design started to delay projects. It was noticed that the software development method that had originally been suitable for mechanics design was no longer appropriate for the company's normal sequential workflow. (Cf. problem 1 above.)

Example 2: A relatively simple software process model based on sequential steps was adopted in a SME (PrE in Table 1) producing complex measuring system for the process industry. It was soon noticed that this model was too restricted for complicated systems design, although, at that point, the development was rather software-intensive. There was no support for the mechanics and optics design, or for pilot deliveries in the selected process model. This case represents problems 1 and 2.

Example 3: The increasing need for better usability changed product development dramatically during the 1990s in consumer electronics and telecommunications. Usability engineering was a new sub-area that did not match any of the typical product development process models, but required cross-engineering and cross-expertise between different organization units. Companies (includes companies from almost all the categories in Table 1) working in this industry had to put a lot of effort into redesigning their processes for usability (Ketola 2002, Jokela 2001). Traditional process models were mostly useless because they did not take into account the specific need of usability prior to the 1990s. This case represents problem 3.

Example 4: A new software house (DE in Table 1) started to produce software tools for web applications. The company adopted development methods and quality assurance practices from its parent company operating mainly in software systems design. The practices were soon found to be too inflexible for "Internet business" in which it was allowed even to release product versions publicly with "unfinished features" or "known bugs". This case represents all the problems 1, 2 and 3.

Example 5: A company (applies to several M company categories in Table 1) was developing software-intensive consumer electronic products, i.e., most of the resources in the product projects were involved in software design. The products, however, were also complex from other design aspects – the development required expertise on industrial, mechanics, electronics, usability design, and effective co-operation between product development, market testing and third-party service providers (i.e. for services working in a final product). In practice, the product was regarded as a complex system, even through it was not a "traditional systems engineering case". The company had problems in applying systems engineering process models, because the work was really software centric, and on the other hand, they had problems in applying a pure software process model because concurrently managed work had so many internal and external connections to other design disciplines and they were not recognized in typical software development processes. This case is a combination of problems 1 and 3.

From the previous problem areas and examples the following general deficiencies for common product development process models can be listed.

The general coverage of a process model is limited: Process models originating from software engineering do not usually pay sufficient attention to other engineering disciplines or business operations in a multi-engineering business area. Likewise, systems engineering originated process models do not sufficiently value software development, which has a major role in many electronic and telecommunication products today. As a result, a large area of business that has a focus on software but also needs expertise in other engineering domains (cf. example 5 above), is left without effective process model support.

Support for user-centred design (UCD) is missing: UCD focusing on good usability and user satisfaction is not recognized by traditional software and systems engineering processes. The design quality of personal electronics cannot be checked using only technology specific quality reviews and inspections that are fundamental for most process models, but the product versions must be validated as a whole against set usability requirements. UCD has a special focus on the concept phase at the beginning of product design process when the correct product features are sought usually together with test users, but it also needs check points and causes possible design modifications in later design phases. If there is a problem in getting software and systems engineering processes to react to UCD needs, there is a challenge in UCD processes to provide guidelines on how to integrate them into existing product development processes efficiently.

The role of early product concepting and specification phases is not recognized: Continuous market and user studies followed by successful product concepting and definition have become important prerequisites for success in the business of personal electronics. When studying the traditional software and systems engineering processes, it can be noticed that they do not recognize this fact. System requirement and software requirement phases have no special role or emphasis in the process. There can also be a conflict if product concepting and specification are part of the development process. However, recognizing the specific needs of the business area, and the increasing role and success of UCD, it is clear that companies must integrate these early design phases more tightly into their product development, either as a clearly defined part of the development process itself, or as a separate process co-operating actively with the product development process.

Process model is not adaptive and flexible: In general, many of the process models are presented highly accurately and at a detailed level by their proposers or inventors. This "accuracy" can lead to problems especially in progressive business areas if companies cannot simplify the model to a relatively simple process guidance. Excessively detailed process descriptions mean that processes cannot be adapted easily to changes in business circumstances that may be due to technology, organizational or market situation changes. The problems of excessively detailed, stable, and inflexible process models are also mentioned in (Cooper 1994, Kamath and Liker 1994, Loch and Kayser 1998, Nihtilä 1996).

Process model does not support customization: Customization of product development processes is not adequately emphasized in the models. This means that many organizations end up attempting to push some model through for their organizations. This may be possible for new growing companies, but forcing a large organization to commit to an incorrect model is difficult and usually unsuccessful. In fact, it should be borne in mind that a process model always needs customization for a specific organization. A good example of process customization is given in (Heikkinen 1997).

There is no support for concurrency: Real support for concurrency between different design disciplines and other company operations is missing or working ineffectively, as both the software and the systems engineering approach fails to define the links from one to the other. There are process model versions emphasizing concurrency inside both approaches, but synchronizing mechanisms between software and systems design views are usually missing. For example, there are no adequate methods available that would provide integrative presentations of the development target and its status during the process, or there are no defined check points for integrative results.

The role of simulations and prototypes is not understood: The progress in technology has led to a situation in which simulations and prototypes are used actively in many process phases and development tasks. Often they are benefited in all product life cycle phases. This enormous potential of simulations and prototypes is not usually recognized by process models. They and their benefits can be mentioned in certain areas or process phases, but there are no process models that value them as an integrative communication means that is used throughout the entire product life cycle alongside the evolution of the real product. Additionally, the models do not consider how to use multi-disciplinary prototypes, how to build simulations and prototypes that are affordable, or how to reuse them and their components.

Support for distributed work and virtual organizations is missing or undefined: Process models do not usually take a strong position towards organization models. This is positive in the sense that it allows more general models to be introduced. However, from the process model customization point of view, it may also involve some negative aspects as many important possibilities to support distributed work may remain unnoticed in the customization work. The new ways of working based on distributed development teams and virtual organizations including partner and subcontractor networks represent such great changes in business culture that their impact should also be visible in product development process models. Although some work describing the development processes in the context of distributed work have been presented both in the software engineering field, e.g. (Aoyama 1996, Rahikkala 2000), and the systems engineering area, e.g. (Pahng et al. 1998, Park and Cutkosky 1999), there is still a lack of process models supporting distributed work in multi-technology software-intensive business areas.

4.1.2 Limitations in tools

Computer-aided tools (see Chapter 3.3) for product development represent a large group of separate tools that are of great benefit in many different engineering tasks. The tools are used in everyday tasks and have improved the productivity of product development dramatically during the 1990s. However, some clear limitations can be found in the features and general use of the tools that should be paid attention to in order to enhance their usefulness. Based on case studies, the following summary can be given of the main problems with the current design tools:

Separation of domains: Separation of the tools into software engineering and other more traditional engineering tools is a sad fact from the point of view of a software-intensive multi-technology company, because it leads to several problems. One real example of this separation is the parallelism of PDM and SCM systems. Basically, existing SCM tools are focused on controlling software items, whereas PDM tools generally concentrate on mechanical parts. From a company's point of view, in most cases there is no need to separate software parts from other product data, but, essentially, clear advantages would be reached if the relationships between mechanical, electronics and software parts could also be expressed. For example, in mobile phones different software versions are needed as the physical UIs in phone models differ in the number of keys, and in size and even the number of the screens. To be able to maintain these links between software versions and mechanical configurations, integrated product data management, or good co-operation between the existing SCM and PDM systems would be needed. Similarly, new integrative tools are needed when personal electronics is designed in the early concepting and industrial design phases, as there is a need to integrate the traditional design, i.e. forms and colours of the mechanical parts, to software controlled UI logic and graphics displayed on the screen. This experience comes from several M-type companies defined in Table 1. The screen in many personal devices, such as PDA's and smart phones, is already a dominant part of the physical appearance of the device. A third example of the separation is the inability of the commercial tools to build heterogeneous product simulations in which the outcome of different design tools could be integrated with a reasonable amount of effort.

Co-operation of the tools: The available tools do not provide open interfaces for connecting tools together from other application areas or from other manufacturers in order to support a fluent work process. There are no adequate standards for cross-engineering information exchange at the time being. In some case companies (ME, HE and MU in Table 1), it was noticed that this lack of inter-operability caused an increase in the number of self-made or shareware utility tools that as an unofficial and non-documented tools caused problems for IT, project and process responsibles. For example, when personnel changes occurred, some small but important self-made tools were left in use without support from their creator. Further development of these tools was difficult if there was no official documentation available. In addition, design tools often fail to take into account the needs of other company operations. For example, marketing and sales need functional product simulations for web pages (experience from company types ME, MU, HE, HU and PE in Table 1). In order to simulate the logic and behaviour of the personal electronic device there is also the need to model the logic and behaviour with some web authoring tool, such as Macromedia Flash. This causes double work as logic and behaviour are, in fact, already modelled in the specification and software design phases. There only should be a means or a tool that would allow employing the already existing material in marketing and sales.

Coverage of the tools: There are no tools to support entirely new engineering fields and product features, nor can the existing tools easily be adjusted for new tasks. For example, mobile service testing and usability tests in telecommunications need a new kind of simulation and testing tools compared to those presented in CAD, CASE and other generic design tools (experience from company types ME, MA and MU in Table 1).

Customization of the tools: Since process models should be customizable, it is only logical that tools should also be customizable to the needs of their owner organization and users. Customization should enable the adaptation of the tools to specific organization models and structures, specific business areas, used technologies, selected process models and user needs. The requirement for customization sets high demands for tools, their implementation technology and architecture. They should be open to support both new extensions for new design areas, and the integration with existing tools. On the other hand, openness and flexibility should be manageable, i.e., there should be adequate tools and guidelines for customization. Excessively open platforms allowing either uncontrolled customization or leading to massive customization projects can become burdensome for an organization. Openness and flexibility of tools were mentioned as limitations of several tools by most of the case companies included in Table 1.

In the next chapter, some specific problems that are related to prototyping tools are covered in more detail.

4.1.3 Limitations in prototyping

Modern computer technology enables the creation of accurate prototypes and simulations for different needs of product development. However, when looking at the process of creating the prototypes and simulations, and their effective use in the product development process, several problem areas can be found. Many of these problems can be directly ascribed to the more general tool problems presented in the previous chapter. For example, the following questions can be posed:

- Is it possible to build a prototype that integrates different technology aspects into one functional presentation? This problem relates to the 'separation of domains' problem above. Many prototypes or simulations concentrate only on some selected technologies or characteristics, such as those presented in the mechanical or software prototypes. To be able to use a prototype as a supporting means for decisions concerning the total product quality, functionality, experience and desirability, the aspects of these separated prototyping approaches should be integratable into one functional presentation.
- Can different prototyping tools be integrated with each other as well as other tools? Can one integrate, e.g., software simulations such as UI

simulations to mechanical simulations? These questions refer to the 'separation of domains' and 'co-operation of the tools' problems above.

- Can our simulations and prototypes be used for new technologies and design tasks? What kind of simulations are needed, e.g., for usability testing of mobile devices and services? This links to the 'coverage of the tools' and 'customization of the tools' problem above.

In addition to generic tool problems, three prototyping and simulation specific problem areas are recognized.

Support for heterogeneous prototypes: As given above, a prototype or a simulation is often targeted only to a certain limited purpose, and thus is does not give an overall product impression. This approach has a background in the problem areas 'separation of domains' and 'co-operation of the tools' described above. The presentation, instead, should aim at cross-discipline simulation integration regarding not only different technology areas but also the parts and components at different maturity levels. This heterogeneous prototyping approach would offer clear benefits as product ideas could be realized faster by using not only simulated parts but also existing products, parts and components. For example, this approach would enable testing and demonstrating a new hand-held fitness monitor by simulating part of the device SW and to use partially existing SW algorithms and hardware.

Support for incremental development and total life cycle: Prototypes are usually built only for specific product development or business tasks. A prototype or a simulation created for a certain task cannot usually be evolved to satisfy the needs of the next task, i.e., its material or components cannot be reused effectively. For example, a prototype built for concepting cannot be benefited in specification, or a simulation created as a part of the specification cannot be used as a starting point for software implementation. The origin of the problem is strongly related to the general tool problem areas 'separation of domains', 'co-operation of the tools' and 'coverage of the tools' that are likely to encourage the creation of isolated, task and technology specific simulations and prototypes. For companies, this approach means a lot of throw-away prototypes and wasted resources and time.

Prototype and component reuse: The problem presented above has to do with effective use of simulations and prototypes within the development and life cycle of different product versions. This view can be extended further to how the creation and use of simulations and prototypes could be better supported over different products, product families and platforms. The challenge here is the reuse of simulations and prototypes, and their components. Personal electronic products are often released as a part of product families and over technology platforms sharing several or even most of the features with other products of the family. Regarding these products, simulation and prototyping tools could also benefit from common simulation templates involving a set of base components to start working with. Platform thinking applied to these areas would facilitate the creation of prototypes and simulations, encourage their effective use over the product life cycle, and reduce the cost and time needed for their creation.

The problems related to the typical process models, tools and prototyping approaches are summarized in Table 3.

Table 3. Summary of the general problems related to process models, tools and prototyping approaches.

Problems in process models (PMP = Process Model Problem)	
PMP1	Process model is too limited to the selected design or engineering areas and does not recognize the multi-disciplinary nature of electronic products.
PMP2	Process model does not include support for user-centred design.
PMP3	Process model does not recognize the importance of early product concepting and specification phases.
PMP3	Process model is not adaptive and flexible enough.
PMP4	Process model does not support process customization to changing business situations.
PMP5	Process model does not have support for concurrent operations and tasks.
PMP6	Process model does not benefit prototypes and simulations adequately.
PMP7	Process model does not recognize the needs of distributed work and virtual organizations that are typical in international business.
Problems in current design tools (TP = Tool Problem)	
TP1	Tools are too specific for selected design areas and cannot integrate design areas into a common view.
TP2	Tools cannot co-operate, i.e., there is no good way to share information between them or to transfer information from one tool to another.
TP3	Commercial tools do not cover the newest design areas.
TP4	Tools are not customizable to specific company needs.
Problems in prototyping (Prototype Problem = PP). In addition to general tool problems TP1 – TP4.	
PP1	Prototypes do not support heterogeneous prototypes.
PP2	Prototypes do not support incremental development and total life cycle.
PP3	Prototypes do not support prototype and component reuse.

4.2 General objectives for product development

The main goal of product development is to support the company's business, i.e., to make the business more successful in the context of the accepted strategy. The role of product development in this goal is dependent on such factors as company size and evolution phase. In general the main challenge in product development is to **be effective** and to support this goal with the following basic sub-objectives that apply to most companies:

1. **Increase customer satisfaction**: User-centred design fulfilling real user needs and preferences is one of the key competitive factors in the electronics and telecommunications industries.
2. **Provide better product quality**: The value of good product quality should not be underestimated in any circumstances. Even product launches introducing outstanding product features can fail if they face quality problems.
3. **Shorten product development times**: Rapid development of new products may not be a key issue as a market demand in all areas of the electronics industry. However, if we understand shortened development times as savings in effort, and thus in costs, the objective applies to most companies.
4. **Support global operations:** New working methods and tools should enable efficient communication and co-operation in global operations and markets.
5. **Support new organization and business models**: Product development must allow integration and adaptation to changing business and organization models. Today, it should be able to act as a functional part of electronic business (e-business) and possibly virtual organization models, but in the future it should be able to support any approaches to which a company will commit itself.

4.3 Requirements for a better development process

By analyzing the problems presented in Chapter 4.1.1 and taking into account the previous general objectives, the requirements for implementing better product development can be stated.

Early validation of product concepts, versions and features: This helps in achieving the objectives 1, 2 and 3 presented above, resolving the problems PMP2 and PMP3 in Table 3. Better customer satisfaction can be attained if design alternatives can be validated efficiently already in the onset of the design cycle. It is realistic to expect better product quality and shorter development times if the amount of costly mistakes can be reduced by way of early validation. In practice, early validation can be supported by heterogeneous modelling, functional product models and effective component reuse:

- Heterogeneous modelling: Heterogeneous modelling refers to the approach presented with heterogeneous prototyping. It means that different parts of a system or a product can be presented with a different abstraction (i.e. modelling, simulation or prototype) level. The goal is to build a good simulation model of a target object that can be used in its early validation.
- Functional product models: When the functions and operations of the target are to be validated, executable and functional models, simulations and prototypes are needed. Design graphs and programming code may be adequate presentations for justifying the quality of the functions for a software engineer, but for a customer or a manager, a more realistic presentation or experience is needed.
- Component reuse: By way of using existing product versions, parts and components, new concepts can be built faster and with less effort.

Efficient communication: The process model should support effective communication at both technical and human levels (see Chapter 3.1). In practice, this means that the technical product development teams must be able to communicate among themselves, with teams of other company operations and with partners and customers. This refers to the objectives 1 to 4 in Chapter 4.2, and PMP 1 in Table 3. The realization of effective communication in the product development process can be assisted by providing total product and process visibility, and by using functional product models:

- Total product and process visibility: The state of the development work and design target should be visible for all of the involved partners in an understandable form at all times.

- Functional product models: Product presentations using basic visualisation means such as images, animations and static models, is the first step for better communication, but making the images or models functional will further enhance the understanding of the product features and work provided by other design team members.

Cross-domain integration support: Product development should recognize and point out tasks and phases that are integrative, i.e., tasks and phases that contain, process or deliver results from or to several expertise areas. Not only should links between the different engineering domains be explicitly stated, but interactions between product development and other business operations should be taken into account. For example, it is obvious that product development and marketing people have to understand each other well in highly competitive consumer markets in order to produce desirable products. Cross-domain integration support has a strong connection with effective communication (see above), and it refers to PMP 1 in Table 3.

Concurrent operations: Concurrency is of benefit not only in specific engineering sub-processes, like in software, electronics or mechanical design, but also between them and between product development and other company functions. For example, market tests, proto series manufacturing can be started in parallel with detailed product design. Support for concurrency means that process model must offer ways for integrating different design disciplines already in the early phases of the product development cycle, i.e., the interfaces between different technology areas in a product must be able to be defined as soon as possible. Concurrency has a major impact on the objective of shortening product development times (Chapter 4.2) and it relates to problem PMP5 in Table 3. In practice, concurrency over engineering disciplines is supported by:

- Total product and process visibly: This is fundamental for concurrency. In other words, the process should offer open views of the overall status of the design target, showing the interfaces between the different tasks and design areas along with their maturity level. Already in the concept design phase, the selected concepts are opened for the design groups, so that they will immediately be able to see what their part in the product is.

- Heterogeneous modelling: Heterogeneous modelling means that some yet unfinished design areas can be replaced with prototypes or simulations, and other design teams can continue their work with the help of these prototypes or simulations. There will thus be no open areas that would hold back some design teams from proceeding.
- Functional product models help especially in validation of design results from concurrent sub-processes.
- Component reuse supports concurrency by providing pre-tested parts of different engineering areas for design teams. In practice, component reuse serves almost the same purpose as heterogeneous models: open design areas can be filled with earlier substitutes.

Component and material reuse: As a major part of components and other material, such as documents, in new products are inherited from previous product versions, the development process should be able to provide explicit support for component reuse. An efficient component reuse contributes especially to objectives 2 and 3 in Chapter 4.2. Pre-tested and pre-used components are normally better in quality, and their use decreases the amount of work and as such shortens development times. Traditionally, component reuse is thought of as a product development operation. However, along the advancement of modern IT tools and networking technologies, new possibilities are also opening for component and material reuse: the product development outcome can now be utilized in several non-product-development operations, such as marketing (paper prints and web demonstrations) and customer support (printed manuals and demonstrative web animations).

Adaptivity and flexibility: There are two main types of changes that are often faced in the product development of rapidly evolving technology and business areas. First, the product specifications may change during the development process. Second, technological, methodological or organizational changes may occur. The aim is to have a development process that can adapt to at least most of these changes without considerable restructuring or redesigning. The development process model should be flexible and not too strict in order to be able to adopt new tools, methods and technologies. The development process should also allow changes in organizational and business strategies. In order to be prepared for these possible changes, a company should aim at (cf. PMP3 and 4 in Table 3):

- Tool and technology independence: It is dangerous to build working methods and processes that are too tightly defined or tied to restrictive tools and techniques. Investments in special tools which do not use open or standardized interfaces are likely to become expensive for a company when changes are needed.
- Iterative and incremental development: Iteration means flexibility and adaptivity of actions, which allows changing circumstances of a business to be reacted to rapidly. For example, changes in design technology or product specifications can be taken into account before moving on to a new design cycle. Likewise, more accurate design decisions can be made based on the results achieved in the previous cycle.
- Process customization: Every organization has its own special needs and features, and all process models need customization for these. Although customization and changes in the process model cannot be avoided, they can be made easier by using a flexible and adaptive base process model.

Distributed product development: The process model, if possible, should explicitly define the operations and phases that can be performed by distributed teams (cf. PMP 7 in Table 3). At a minimum level, the tasks that cannot be distributed should be recognized and marked in the model. The realization of the work distribution is supported by a component-based product structure, functional digital product models and components, and heterogeneous modelling:

- Functional, digital product models: By using functional digital product models, it is possible for distributed teams to efficiently test their work results together with other product parts. There is no need for physical parts and as such the co-operation can be performed through the Internet and local area networks (LAN). So, the interfaces between components and the functionality of sub-parts and the entire product versions can be verified in advance before the physical parts are constructed.
- Heterogeneous modelling: Heterogeneous modelling further supports distributed work by providing an even more extensive selection of choices for building functional product models and their simulations that work over data networks.

Utilization of modern IT technology and tools: This requirement is fundamental for efficient product development and acts as a default demand for every other requirement mentioned above. This means that advanced simulation, prototyping, networking, design and other IT tools are needed in order to run the development work, and the possibilities of different tools must be acknowledged and stated in different process tasks and phases.

4.4 Requirements for better tool support

Based on the problems presented in Chapters 4.1.2, a set of general requirements for better design and design support tools can be given:

Flexible interfaces: Tools should use standardized interfaces when accessing data or interfacing with other tools. If no standardized or de-facto standard interfaces exist in the area, tools should provide flexible and open interface components that allow the customization of the tool to a different data format and other tools. This requirement refers to TP1, 2 and 4 in Table 3.

Support for customization: If possible, tools should be customizable to new tasks and technologies. They should have some programmable capabilities that allow the building of new features to tools. With this possibility, tools can help in covering, at least partly, totally new design tasks, before specific tools are developed. In addition, customization should enable the down-scaling of the properties so that a suitable feature set of a tool can be found for various organization sizes. (Cf. TP3 and 4 in Table 3.)

Standard or de-facto base technology: Tools should be based on standardized or otherwise mainstream base technologies. This requirement ensures that the tools have more additional tools to interface and co-operate with. Recognizing this requirement can help avoiding the problems TP1, 2 and 4 in Table 3.

Flexible architecture and technology base: Tools should be flexible in their architecture and base technologies. This means that they should be easily adaptable to different technologies, design methods and notations without extensive update needs. As an example, a property to support several modelling formalisms in software logic modelling can be mentioned. With this approach, a tool is not tied to support only one technology, design method or notation, and it can thus better serve for meeting unpredictable changes in them. This requirement relates especially to TP3 in Table 3.

Maintaining product and process related information: As different tools provide a view of the target product from a certain perspective, they should take the full benefit of this expertise and collect, store and manage all of the information concerning the design object, its components and design status. This information can be of great benefit for different stakeholders in different product life cycle phase. This requirement is directly linked with especially TP1 and TP2 in Table 3.

4.5 Requirements for better prototyping and simulation

The requirements for better prototypes and tools supporting their creation and use can be concluded from the problems presented in Chapters 4.1.2 and 4.1.3.

Flexible and customized presentation of the design target and information related to it (i.e. utilization of product and process information): Prototyping tools should take into account the different product life cycle phases, business and product development process phases, along with other business operations. This requirement refers to the general tool problem TP1 and the prototyping specific problem PP2 in Table 3. Tools should be able to present product models and related information according to the needs of specific tasks, people and process phases:

- *Support for different views of the target object and it status:* Design team members and other company employees should have the needed presentation of the model with the necessary information, i.e. different views of the product information. Tools should support the access and presentation of varied data. Different types of UIs, controls or functions should be attachable to the models according to the target task. There should be libraries of components for specific tasks and process phases.

- *Different levels of detail:* The level of detail in simulations should be variable, i.e., small-sized models could be derived for web use, while accurate simulation could be used in detailed product design.

Support for cross-engineering product presentation (cf. TP1 and 2 in Table 3): Prototyping tools should support an adequately accurate and realistic presentation of the target product. Different features should be able to be emphasized according to their application, while, in general, software, electronics and mechanics should be able to be combined in the tools. The presentation should be realizable with a degree of realism that allows either the customer or the company to justify the quality and the value of the product. The following dimensions of the product should be describable:

- *Visual appearance of the product.* The shape, materials and colours of the target should resemble the original source. The appearance must also include the elements seen in the software-based GUI. Even a photorealistic presentation of the object should be possible, especially in certain tasks with direct customer contact, such as marketing and customer support. Taking into account the visual appearance of the product also supports the experts working in areas such as industrial design, mechanical design, graphical UI design, usability and marketing.
- *Functionality of the UI*. The UI of the product including displays and their content, lights and mechanical controls, such as push buttons and sliders, should be presentable with such a realism that the user can obtain a realistic feeling about the usage situation. Functional UI helps in integrating the views from mechanical and industrial design into UI graphics and software design, and thus helps in concepting, specification and usability testing.
- *Functionality and behaviour of the product*. The logical and mechanical behaviour of the product should be realizable so that the operations could be tested and experienced realistically. If needed, the prototyping of a product's behaviour should also support the integration of external devices and services to the target product. Prototyping for product functions and behaviour is of benefit especially in detailed product specification and software design. It can extend the approach from a higher-level UI prototyping to a more detailed design level also taking into account the software architecture and implementation questions.

- *Audio properties*. Different sounds, tones, music and voices attached to the object or its functions should be able to be realized. An ability to prototype audio properties helps experts to have a more concrete view of the target product, e.g., in concepting, user testing, audio design and marketing.

Versatile import and link capabilities from different design tools: Prototyping tools should be able to import models and other components from various designs and IT systems. Models and components should be importable through an intermediate format or with the help of format specific converters. Other components, such as sounds and logic, should be able to be incorporated into the model from common CAE or design tools, or open interfaces to these should be offered. Fulfilling this requirement can help in avoiding problems TP1, TP2 and PP2 in Table 3.

Versatile export and link capabilities to different design tools and IT systems: Prototypes and their components should be exportable or linkable to other tools, such as basic office and web tools, and specialized tools, such as software development environments, database and product data management (PDM) systems. This requirement, too, has a direct impact on TP1, TP2 and PP2 in Table 3.

Support for heterogeneous prototyping (cf. PP1 in Table 3): Prototyping tools should be able to integrate different technology areas, as well as parts and components from different maturity levels, also including real products and services.

Support for incremental prototyping (cf. PP2 in Table 3): Prototyping tools should support incremental use of prototypes over different tasks and product life cycle phases. Simple and basic prototypes should be expandable step-by-step towards more advanced and complex prototypes, thus approaching the real target product in terms of features and accuracy of presentation.

Support for prototype and component reuse (cf. PP2 in Table 3): Prototyping tools should not only support prototype reuse within one product model or development project (cf. incremental view below), but over several product models, product families and platforms. Additionally, the tools should encourage component reuse when building new prototypes for different purposes. The selection of component libraries and template projects has a major role in putting this requirement into practice.

The requirements for better product development practices are summarized in Table 4.

Table 4. Summary of the requirements for better product development practices.

Requirements for a better product development process (RDP = Requirement for a better Development Process)	
RDP 1	Supports early validation of product concepts, versions and features.
RDP 2	Supports efficient communication.
RDP 3	Cross-domain integration support.
RDP 4	Supports concurrent operations.
RDP 5	Supports component and document reuse.
RDP 6	Supports adaptivity and flexibility.
RDP 7	Supports distributed product development.
RDP 8	Supports utilization of modern IT technology and tools.
Requirements for better product tool support (RTS = Requirement for better Tool Support)	
RTS 1	Offers flexible interfaces.
RTS 2	Enables tool customization.
RTS 3	Uses standard or de-facto base technology.
RTS 4	Offers flexible architecture and technology base.
RTS 5	Maintains product and process related information.
Requirements for better prototyping support (RPS = Requirement for better Prototyping Support)	
RPS 1	Supports a flexible and customized presentation of the design target and information related to it (i.e. utilization of product and process information)
RPS 2	Supports the cross-engineering presentation through an integrative prototype including: – Visual appearance – Functionality of the UI – General behaviour – Audio properties.
RPS 3	Offers versatile import and link capabilities from different design tools.
RPS 4	Offers versatile export and link capabilities to different design tools and IT systems.
RPS 5	Supports heterogeneous prototyping.
RPS 6	Support for incremental prototyping.
RPS 7	Support for prototype and component reuse.

5. Virtual design

This chapter defines the term virtual design and explains the rationale behind it. First, virtual prototyping is introduced and defined as a fundamental element of virtual design. After this the objectives of the virtual design approach are identified and the basic process model for its implementation is introduced. The model execution is illustrated and explained step by step at a general level. Next, another perspective of virtual design is provided from a product life cycle support point of view. It is also shown how virtual prototype instances used in virtual design can benefit the company and its customers in the different phases of product life cycle.

5.1 Virtual prototyping

Virtual prototyping can be viewed either as a technology term or as a process description. When considering it as a method of working, it can be defined as:

> ***Virtual prototyping*** *is a process in which a product or a product concept, its behaviour and usage are simulated as realistically as possible by combining different prototyping and simulation models and techniques.*

If a technological approach to virtual prototyping is taken, the focus will usually be on the virtual prototype and its realization:

> ***A virtual prototype*** *is a simulation of a target object which aims at an adequate degree of realism that is comparable with the physical and logical functionality and the appearance of the possible real object, and that is achieved by combining different prototyping and simulation models and techniques.*

The presented definition of a virtual prototype is based on the definition given by Haug et al. (1993). However, there are some significant differences in the expressions. First, in our view of virtual prototype, the simulation of functionality and appearance are both needed. Secondly, in the realization of a virtual prototype our approach emphasizes the prototyping of the software and software implemented product features, not only the mechanical parts and their design as is typical in Haug's work.

In general, the implementation of virtual prototypes does not rely solely on computer models but also on semi-digital ones, i.e., partial hardware mock-ups may be utilized in the process. Finally, in virtual prototyping, the aim is to provide a multi-dimensional view of the target object. The focus of interest is not restricted to a single or a few characteristics of the object, such as physical dimensions, but the idea is to provide as much useful information about the target as possible so that the user or customer can examine the total quality and the features of the target from the impression given by the virtual reality prototype.

Virtual prototyping integrates several techniques and expertise areas, such as advanced modelling techniques, multi-disciplinary simulation and prototyping techniques, interactive UI design and software development. Virtual prototyping also supports heterogeneous prototyping, but in a wider sense, as understood in software engineering (Pulli and Elmstrøm 1993, Gabriel 1989). According to the view of software engineering, heterogeneous refers to the different abstraction levels of software rather than the abstraction levels of the entire product design, which also include mechanics, design, electronics, et cetera. Based on the previous facts, it can be seen that virtual prototyping clearly fulfils the requirements of RPS 2 and RPS 5 as given in Table 4.

Although prototyping systems that combine different design areas have been introduced, such as in the area of co-simulation (see Chapter 3.4.4), the need is obvious for a comprehensive approach supporting all of the participating engineering aspects and covering the entire development process from the acquisition of user preferences to product marketing. Virtual prototyping is a step towards this goal. When studying Table 2 presented in Chapter 3.4.3, it can be noticed that virtual prototyping, in fact, functions as an umbrella for all of the other approaches. It can explicitly cover all of the goals set by the other approaches, except for rapid prototyping. However, rapid prototyping can also be combined with the virtual prototyping process, either as a part of semi-digital prototypes or as a subsequent process phase for which virtual prototyping is utilized as a pre-test in order to ensure the design results prior to building rapid prototypes.

An advanced human-computer interface (HCI) is one of the basic elements in virtual prototyping. It can be implemented as a computer simulation controlled with traditional input and output devices, such as a display, mouse and keypad. Another possibility is to use hardware mock-ups or real devices, such as an existing mobile phone, to provide an interface to simulated features. Also VR techniques can be used to provide a strong intuitive sense of reality and the means to change the simulation through meaningful interaction. When VR techniques are applied in virtual prototyping the approach can also be called virtual reality prototyping (VRP). VR can have an important role, especially, when large-scale products and systems, such as cars and airplanes, and their environments are prototyped. More about the implementation architectures of VRP can be found in, e.g., (Salmela and Pulli 1997, Kerttula et al. 1997b).

5.1.1 Simulated features

In virtual prototyping, the typically simulated features include (Figure 9):

- Visual appearance of the product including the shape, mechanical parts, textures, surface materials and colours of the target. The simulation of these features is inherited from techniques used in mechanical engineering and industrial design. The weighting of these features varies according to the needs set for the simulation. For example, in concept design, the shape is usually the most important feature, while in marketing the importance of the textures and colours increases.
- Functionality of the physical UI, including displays, lights and mechanical controls, such as push buttons and sliders.
- Internal functionality and behaviour including the logical and mechanical behaviour of the product. With software-intensive products, the role of software implemented features is emphasized. Their simulation is strongly linked to software development techniques and tools.
- External functionality including connections to and from the outside world. It also includes stimulus and functions that are reacted or accessed though hardware or software, but not directly by the user. For example, mobile services, software agents and sensory input in personal electronic products represent these simulated features.

- Audio properties including different sounds, tones, music and voices attached to the object or its functions.

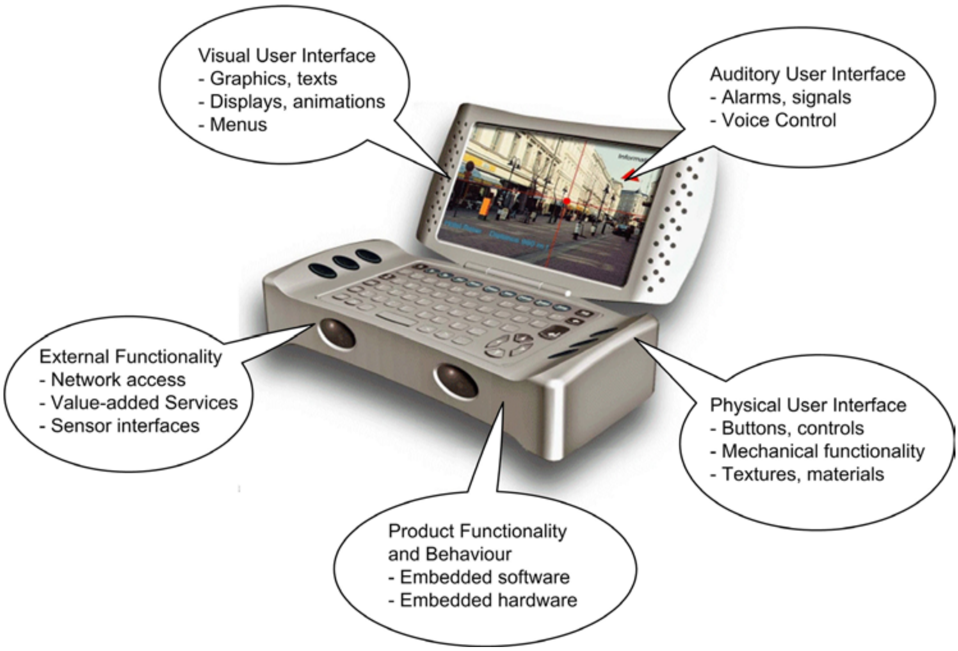


Figure 9. Virtual Prototyping – Simulated Product Features
(CyPhone model courtesy of Metsävainio Design Oy, Finland).

In addition to the simulation of basic product features, a virtual prototype presentation can be expanded with other information that is usable in certain situations. For example, assembly information can be added to virtual prototypes in order to help technicians in field installations. Similarly, additional text can be included in a prototype presentation, e.g., when the presentation is to be used as product specification explaining the functioning of a mobile application in a specific phone model. Sometimes it is necessary to prototype or visualize a complete usage situation in advance. It is then that prototypes can be integrated to form a complete presentation of a new product idea. For example, prototypes of different devices can be synchronized together to form a distributed system prototyping system. If VRP is applied, even more realistic and advanced simulations of usage situations can be built. In VRP a more extensive simulation experience can be reached with desktop VR arrangements (Kalawsky 1994), in which a virtual working space can be created between the user and a computer screen by using VR devices. Desktop simulation can already provide an immersive simulation experience, which is sufficient for many purposes, especially if the target is a small hand-held object. A sophisticated desktop simulation can be realized, for example, with the following VR devices:

- stereoscopic system for stereoscopic viewing
- trackers for tracking e.g. head movements, hands or physical objects
- data-glove for manipulating an object, such as a physical mock-up of the product
- 3D auditory system for realistic audio simulation
- haptic device for providing tactile and force feedback of objects.

5.2 Definition of virtual design

To study the process related aspects of virtual prototyping further, i.e., the possibilities of applying virtual prototyping as a new way of working in an organization, the approach is extended to virtual design. In this context we define virtual design as:

> ***Virtual design*** *is a product development and business process framework that applies virtual prototyping and virtual prototypes in selected process phases and tasks.*

Although the term only contains the word *design*, and not explicitly the word *business*, the definition is not restricted only to product development. Depending on the company's organizational model the boundary between product development and other activities may vary. For example, in some companies marketing or a part of it may be included in product development. This is reasonable, especially if the marketing people are working closely with product developers in activities such as product definition, concept design and usability testing. In addition, in the high technology industry product development often has a central role in an organization, and other business activities are built around it to extend its operations, so as to provide for a profitable business. Of course, this role of product development will often change as the company and its business mature. In general, it can be stated that product development in a modern company has or should have versatile links to and active co-operation with other business operations of the company. Product development cannot be studied as a separate segment in a company. For that reason, it would be impractical to restrict the definition of virtual design to only product development.

Why, then, aren't terms such as 'virtual business process', 'virtual enterprise', or 'virtual organization' used? In general, the situation regarding the terminology used for virtual-based processes and organization is similar to that of prototyping terms discussed in Chapter 3.4, in other words, messy. There are an enormous amount of terms in use, and even more definitions given for them. In this era of globalization and Internet technology, the business world and business economics are going through a hectic period of development, in which no globally accepted definitions have been fixed as of yet for most of the terms. The basis for the definition of virtual design presented here is in virtual prototyping based product development, which is why the emphasis is on technology and processes, and not on organization structure, to which many of the previous terms would refer to. For example, virtual enterprising, as defined by ESoCE (European Society of Concurrent Engineering, Paris, France) in (Pallot 1998), refers to the virtuality of a business organization, and not so much to the virtuality of the design target, design technology or design process.

Virtual design can be understood as a product development and business process framework that applies virtual prototypes through the entire product life cycle. The idea of the process is to create a shared product presentation with a virtual design space allowing all of the stakeholders can experience the latest status of the target product. Contributing participants, such as product developers, develop the presentation further, i.e., provide new material and components to the presentations. On the other hand, the benefiting parties such as marketing people utilize the presentations in selected forms in their own tasks.

Virtual design would not be truly worth consideration in companies without the increasing role of software in the products. It is software that ties virtual design to reality. When designing software-intensive products, the border between virtual and real product components becomes fuzzy. A virtual presentation of a product, indeed, can contain real product components. In fact, the major part of the design target can consist of real product components or so realistically simulated product components that the overall virtual product presentation is usable for several needs. This allows important product decisions to be made earlier and material from the product presentation to be adapted earlier in different operations. For example, if the UI of the presentation equals the final product, the virtual presentation can be used as such in web-based training.

5.3 Related work

Virtual design defined as a complete product development and business process model is a relatively new phenomenon in the research world. There are some references to virtual design and the term is used in articles, but usually these refer to design practices related to virtual prototyping (see Chapter 3.4.4) or VR (see Chapter 3.4.2) technologies. For example, Noor and Ellis (1996) introduce engineering in virtual environments and offer MUSE and CAVE as examples of environments for virtual design. Often virtual design also shares the characteristics of collaborative virtual environments aiming toward a CSCW. Distributed and collaborative virtual prototyping are presented, for example, in (Jasnoch and Anderson 1996, Lombardo et al. 1996) as well as in (Horvath and Rusak 2001). In the context of the previous design areas and tasks many synonyms, such as virtual product design, virtual product development, virtual prototyping environment and virtual engineering, can be used instead of the term virtual design. Sometimes virtuality of the terms also refers to the organizational aspects of the design work. For instance, virtual engineering can mean engineering work performed in a distributed organization.

Direct references to virtual design are given, for example, by Dani et al. (1994), who describe a virtual design system for conceptual design. COVIRDS (COnceptual VIRtual Design System) enables the design of new product concepts in 3D by using a combination of hand gestures, voice input and keyboard (Dani et al. 1994). Another virtual design system was introduced by Lau et al. (2003). The authors present an Interactive Design Visualization System (IDVS) that supports web-enabled engineering by visualizing product models as 3D VRML models in an Internet browser (Lau et al. 2003). The users, such as designers and customers, can access the model from different locations and provide comments synchronously by using the CSCW features of the system. A virtual design research that has similarities with the approach presented in this thesis is introduced by Tseng et al. (1997). They present a virtual design framework for product customization, which aims at a multi-disciplinary definition of a product at the product family level. Tseng et al. (1997) emphasize the role of the customer in the design process and offer immersive prototypes as a means for early understanding of product features. Despite some similarities with the virtual design presented in this thesis, Tseng et al. focus only on design and manufacturing simulation of mechanical products. They do not address or recognize the software and electronic parts of the products.

5.4 Objectives

Virtual design is a process framework that tackles the problems encountered with current design processes, tools and prototyping approaches as described in Chapter 4.1. It also suggests a new kind of working approach for product development, especially now that the overall business structures are in a transitional state, because of globalization and modern IT progress. New business strategies, such as commitment to e-business and CE, and their advantages are hardly achievable if they are not backed by all of the company's activities. Virtual design is a step towards this business restructuring progress.

In practice the aim in virtual design is to combine different design and expertise areas, including industrial design, software, hardware and mechanical design, UI design, and usability testing in a virtual prototyping based process model, and as such to meet the requirements given for a better product development process as was shown in Chapter 4.3 and in Table 4. This approach aims not only at supporting system-level design, detailed design and testing, but also at facilitating and providing means for concept design and marketing activities. The proposed model is to combine advanced prototypes with virtual prototyping techniques in order to create total product visibility for all of the design teams, to support iterative product refinement to the final version of the product, and to enable making use of existing design components and product versions. A further objective is to strengthen customer-centred product development by emphasizing direct customer requirements and preferences.

5.5 Basic process model

Virtual design is demonstrated with the virtual design process model introduced by Kerttula in (Kerttula et al. 1997b) and in (Leppälä et al. 2003). The aim of the model is to function as a reference process for companies applying virtual design. It focuses on a simple presentation easily understood by product developers and other stake-holders having different backgrounds. The virtual design process consists of two main phases: the initial concept design phase and the virtual design phase. During concept design, developers create the first product models and place them into the virtual design space. In the actual virtual design phase, product designers develop the selected concepts to a final product version or versions. The virtual design phase includes five main component types:

- *Virtual design space*. This space forms the core of the design work by integrating design information and components into a common functional and interactive product presentation. It provides the needed views of the design target and its status for process actors. The virtual design space is illustrated in red in Figure 10 describing the virtual design process.
- *Actors*. These elements include development teams, company departments or functions that process information and material, and update virtual design space and its models through design data and a component repository. The actors are emphasized with light green in Figure 10.
- *Information and material flows* show what kind of information or material is transferred between the Actors. These flows are marked with light yellow in Figure 10.
- *Design data and component repository*. The building blocks for the product model include parts produced during the current design process or existing parts recycled from earlier products and projects. The repository content can vary from design documentation to actual product components and they may even include external tools and simulation systems. The repository is given in orange in Figure 10.
- *Design target view and status.* This view emphasizes the total product visibility that the integrated virtual design space and virtual product models offer to design teams and other participating members. The target view is marked with blue in Figure 10.

The process model is presented in Figure 10. The numbers in the figure refer to the main operation of the process. Steps 1 to 5 are performed in sequential order:

1. Design first concepts. The initial concept design is based on existing business knowledge and product versions combined to match new customer needs. If there is no existing material available, the first concepts are produced based on new product ideas and visions. The amount of effort needed for this phase varies considerably, depending on whether a completely new product is being designed or whether an existing product is being updated. The results of this phase are the initial requirements and specifications for the target product and its first 3D industrial and mechanical design models. The players here usually include product marketers, concept designers, usability experts, and the management.
2. Utilize and redefine the concepts. The results from the initial concept phase can already be used in early marketing activities and in obtaining more accurate customer feedback. If suitable tools are available, functional concept models can also be built and feedback for them can be collected from test user groups already in this initial phase.
3. Place concepts into a virtual design space and refine them further. Selected product concepts are placed into the virtual design space. Sometimes even several concepts can be developed to a more mature level prior to the final product decisions. The concept presentations in the virtual design space simulate and visualize the design goals for all participants in a development project. The participants use their own expertise to further specify the concept or concepts for the final product specification. Active communication is needed between different development teams, and with other business units, such as marketing and manufacturing. Various technologies are utilized to present concepts and their features. For example, the first prototypes are built from the initial 3D CAD drawings by adding mechanical functionality to them, and the logical behaviour of the product is defined with software engineering tools supporting rapid logical modelling. These are combined to form a functional product model that is utilized in deciding the final higher-level product specification.

4. Make a product decision. Based on the available concepts and preliminary information related to their implementation costs, market estimates, et cetera, the final product decision is made. Specifications are fixed and team members start working on creating the product for markets. The virtual product presentation in a virtual design space demonstrates and helps in understanding the specification and the goal of the actual product development. Depending on the organization the final product decisions are usually made by the marketing department or general management.
5. Begin detailed system specification and integration. As soon as the product decision has been made, detailed technical specification and design is initiated. System integration can also be launched as soon as the interfaces between the different modules and components are defined. Simulated components are built and used when needed. This enables a faster integration and testing of separate parts.

Steps from 6 to 12 are run as parallel operations.

6. Integrate existing components into the product model. When developing new product variations using existing product components, the target object can be extended rapidly by integrating the existing components into it. For example, a selected UI style with default functions and graphics can be added to the model, or an existing hardware platform can be linked to the new product specification to enable testing of real hardware based operations.

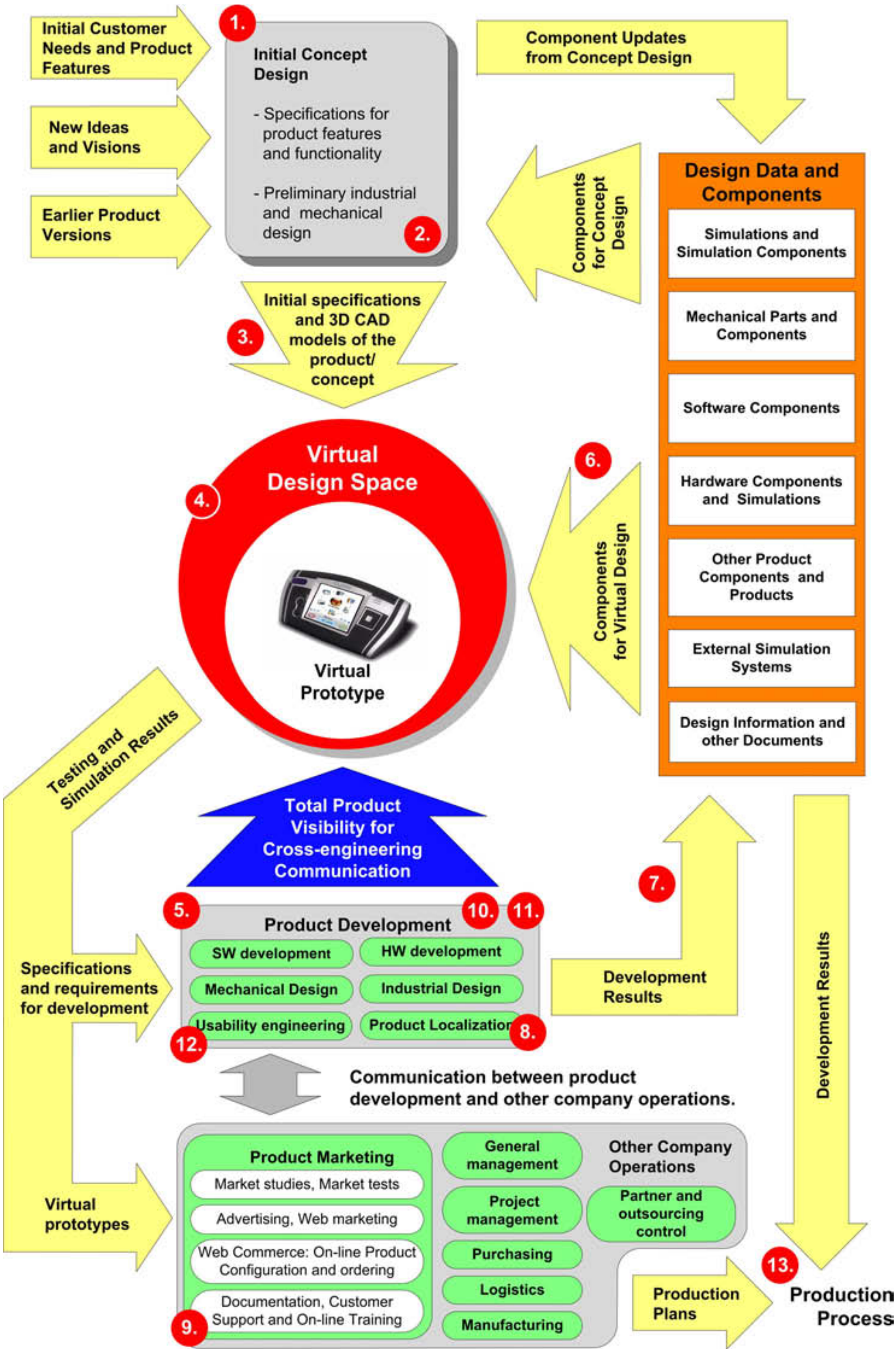


Figure 10. Virtual design process model.

7. Develop new product components, add them to the repository and integrate them into the product model. Development teams produce new components, technical documents and other material for the design data and component repository. They are unit tested and then integrated into the product model. The process model does not restrict the implementation of different design areas, but the detailed design work in various expertise areas and teams can be performed with the most suitable approaches at the time.
8. Localize the product and related material. As soon as the design of selected parts has been conducted in the default language, localization specialists can begin their work in product localization. Localization can include the localization of product components, as display texts, icons, cover prints, et cetera, and product documentation such as user documentation and its images.
9. Produce user documentation, and marketing and customer support material. Technical writers and marketing people can begin their work as soon as the product components are mature enough for their needs. For example, as soon as the product design is fixed, marketing can start the production of printed brochures and electronic marketing material. Technical writers can start compiling user documentation as soon as UI and the main functions are defined and accepted. They can benefit directly from the exact graphical material coming from UI simulations, if the material is accurate enough. In addition, technical writers can receive localized material from product localization. Functional simulations can also be modified to be used in web-based customer support.
10. Perform integration and system testing. These activities can be started as soon as the interfaces between different components and parts are defined. In practice, the work is started a lot earlier than in the traditional development process, because unfinished components can be substituted with prototype parts mimicking the real ones under development. This phase can be boosted by making efficient use of existing components and parts.
11. Integrate and test co-functioning with 3rd party parts and systems. As electronic products offer more and more solutions integrated by several companies, the role of multi-contributor integration and testing is

significant for a successful product launch. Just as in the previous Step 10 the integration and testing of 3rd party components and parts in the design target can be started as soon as the interfaces have been specified. The co-functioning of the design target with external systems and tools can also be started earlier. For example, the testing of mobile network services and 3rd party mobile services with a new mobile terminal model can be performed more reliably already prior to the launch of the terminal model. Note: 3rd party players are not illustrated in the figure as separate actors.

12. Take care of usability. User-centred design is supported by continuous usability testing benefiting from evolving virtual product presentations. The specifications for usability work with a particular design target are defined already in the concept design phases where the overall product positioning is done. The target customer group and user experience for the product and its functions are defined. In the virtual design phase usability engineering focuses on controlling the given overall specifications by continuously refining the product details to fit to them. Usability engineering requires constant communication between usability experts, marketing and design teams. Although the product is defined in the early phases, the specifications hardly remain unchanged through the process, but changes and modifications are needed as the process matures. In these changes usability engineering has an important role in ensuring that the overall product specifications are not violated.

Final phase 13, complete the virtual design process.

13. Make the decision to finish the development and start production. Production can begin once a sufficient number of development cycles have been performed and the manufacturing process has been designed and tested. Depending on the product under development, most of the tests needed to ensure the product performance can be done within the virtual design space, and only manufacturing related tests are needed to be performed during production.

The model structure and its execution sequence presented here should be considered only as a process framework. In the real world, the implementation and realization of a model will always need organization and business dependent modifications.

5.6 Product life cycle support

When looking at virtual design from the business point of view, it is important to recognize the role of virtual prototypes through the entire product life cycle. The life of the virtual product presentation is not restricted only to the development phase, but it also exists parallel to the actual product from the first concepts to the final customer support. This parallel use of a virtual product is beneficial not only for the company but for the customer too (see Table 5). For the customer it means better service from the company, and an increased satisfaction towards that specific company. For a manufacturing company it means better product targeting, a more satisfied customer, internal cost and time savings offered by material reuse. The issues explained in the table apply not only to consumer and industrial product businesses, but also to OEM business. Many of the advantages also apply to partner network co-operation.

Table 5. Virtual product models in different product life cycle phases.

PRODUCT LIFE CYCLE	Product Concepting	Specification and Development	Manufacturing	Marketing and Advertisement	Sales	Product Training	After Sales
Customers viewpoint							
	Customer Feedback	**Selecting**	**Customising/ Creating own**	**Trying**	**Making decision/ Shopping**	**Learning**	**Getting help**
Approach	- Customers can see and try new concepts and visions. - Give your needs for new products.	- Test, try and compare products, even at home via web, in advance.	- Configure or modify your own product from components. - Configure components and features you need. - Propose or add even own product features.	- Try, test, view and compare products, even at home in advance.	- Order and buy product through the Web.	- Get help, support and tips for your product.	- Get support and services for your product. - Get information about accessories and updates available.
Customer Benefits	- Better products in the future.	- Offers more information about products. - Home trials save customers time.	- See the prices for configurations at once.	- Functional products models offers more information about products. - Home trials save customers time.	- Buy in home, get delivered to home. - Save time and costs. - Have easier shopping.	- Better learning through demonstrative product visualisations and animations. - 24 hour service and access via web.	- Access demonstrative support with visualised and animated operation instructions. - 24 hour service and access via web.
Product company's viewpoint							
	Concept design and specification	**Testing/ Presales**	**Production/ Manufacturing**	**Marketing**	**Sales and delivery**	**Training/ User guidance**	**Customer support/ Service**
Approach	- Apply virtual prototypes in concept design. - Test product and collect information about markets, also on the web.	- Apply virtual prototypes in product design. - Test product, advertise and collect information about markets, also on the web.	- Apply virtual prototypes in production testing, assembly training and guidance.	- Market products through dealers and the Web - Advertise and collect information about markets.	- Connect virtual product models to e-commerce systems and sales configurators.	- Provide customer training and help with functional virtual prototypes on web and other electronic media.	- Deliver and demonstrate product updates, accessories and spare parts with virtual prototypes.
Company Benefits	- Better cross-engineering communication internally. - Support for distributed co-operation. - Better customer appeal and satisfaction through continuous customer feedback.	- Exploit the Web: -> Good coverage with standard tools. -> Cost-effective channel. - Apply product material available from R&D: -> Save costs and time.	- Optimize production phases and lines in advance with lower costs - Train workers for new tasks in advance: -> Save costs and time.	- Exploit the Web: -> Good coverage with standard tools. -> Cost-effective channel. - Apply product material directly from R&D: -> Save costs and time.	- Get direct link from user's selections and configurations to sales and delivery: -> Reduces work and saves costs.	- Replace or minimise paper documents. - Web provides easy update -> Save costs and work. - Offer more effective and demonstrative help -> Better customer satisfaction.	- Exploit the Web both as information and delivery channel: -> Save work and time. - Offer more effective and demonstrative info: -> Better customer satisfaction.

6. Tool support for virtual design

Tool support has a significant role in the successful implementation of virtual design. Some of the basic ideas of virtual design can be implemented already with currently available commercial tools, at least, within certain sub-processes or design areas. However, it is difficult to reach the full integrative tool support for virtual design with the current commercial tools because of the problems presented in Chapters 4.1.2 and 4.1.3. The main problems are in the fact that the development tools cannot co-operate across over different design areas, and the prototyping tools do not support multi-disciplinary and heterogeneous presentations. In practice, there are two alternatives to overcome the lack of commercial tools that could implement virtual design. One of these is to build customized tool support by using the available commercial tools and self-made extensions for the integration parts and areas that are not supported otherwise. The other alternative is to wait for a new kinds of commercial tools offering support for virtual design.

In this chapter, it is studied how tool support for virtual design could be implemented. The work is started by analyzing architectural approaches to implementing virtual design space and also design area specific tools contributing to it. Selected commercial tool categories are studied in order to evaluate their possibilities in building tool support for virtual design. The VRP Environment from VTT and Cybelius Maestro tools are examined closer as implementation solutions for virtual design of personal electronics. The experience from and benefits of these tools are analyzed further in design studies presented in Chapter 7.

6.1 Implementation architectures

A natural architecture for implementing a virtual design environment consists of shared virtual design space, and design area specific tools contributing to the design space. A virtual design space[21], basically, is a computer and software system that stores the latest design information and version history of the target product(s). Basic product information can be extended with other additional process and life cycle phase information, including, e.g., information on the status of the parts and development project, and retained quality actions. The main purpose of the virtual design space is to keep all of the product team members aware of the latest status of the product under development. The virtual design space should be able to integrate design results coming from different design areas to a common presentation utilizing virtual prototypes. This presentation, on the other hand, should be exportable to other tasks, such as usability testing, localization and marketing, requiring an integral prototype presentation or its modifications. To be able to support virtual design space, the design area specific tools, i.e., design, design analysis and design support tools mentioned in Chapter 3.3, should have flexible interfaces for exporting their results to the virtual design space.

An alternative choice for the previous architecture using a separate virtual design space is an architecture in which the design space is offered by one of the design tools or tool platforms covering one of the main design areas. This architecture may be limited in some dimension, as all of the design areas may not be supported by this approach. However, it may also be a premeditated act to leave some design areas out of the virtual design space or to let them integrate only loosely into it. In the case of software-intensive personal electronic products, the suitable master tool used for taking care of the design space functions will most probably come from the area of software development or UI prototyping. This is because UI and the software features controlled by the tool have a key role in the business success of these user-focused products. Developing proper interfaces and filters for the selected tool will enable the importation of substantial material from industrial, mechanical and hardware design areas, and thus form a multi-engineering prototype presentation. By extending the master tool further with selected features, the created prototype can be made use of in many life cycle tasks, including concepting, specification, localization, implementation,

---

[21] Virtual design space has been studied also outside the context of virtual design process. For example, Kari Kuutti at al. (1997) studied virtual design space as communication and collaboration enabler.

documentation and marketing. In general, the architectural approach integrating the virtual design space into a selected design tool sets strict requirements for the selected master tool. The master tool should meet most of the requirements listed in Table 4 for tool support (RTS items) and for prototyping support (RPS items). Later on in Chapters 6.3 and 6.4 the VRP Environment and Cybelius Maestro tools are introduced as examples of tool architecture embedding virtual design space functions in a specific design tool.

6.2 Potential of commercial tools

In the next chapters, some promising categories of tools that could be used for the creation of virtual design tool support are studied.

When considering the first architecture choice and the realization of virtual design space, it seems adequate to study collaborative tools. These tools enable communication within the design team or between teams, and they support material sharing asynchronously or synchronously over computer networks. At present, most of the collaborative tools make use of the Web, and can share different kinds of material, ranging from documents to visual content, product components, and so on. The material is stored on databases or network servers and accessed via a special web interface under secured connections.

When considering collaborative tools as an enabler of virtual design space, most of the challenges have to do with how to present and integrate material from different design areas in a collaborative tool. Many of the design tools designed for mechanical and industrial design have their own plug-ins for web presentation. This helps the first issue, but the question remains as to how to integrate software functionality into these CAD presentations. Solutions for integrative product presentations are given especially in collaborative virtual environments (CVE), which make use of VR technologies as a part of collaboration.

Many of the CVEs are, however, still environment focused, i.e., they do not focus on product presentations (Kerttula et al. 1999b), and they provide no support for defining the product’s behaviour with the typical practices used in software engineering. The former means that a single product such as a personal electronic device may be difficult to present with adequate accuracy and with a degree of realism sufficient for detailed product development. The latter means that the modelling of software behaviour for the target product may be done with a proprietary language, such as VRML (Virtual Reality Modelling Language), which does not link naturally to the real implementation of the target product's software. The possible duplicate design effort needed for these separate presentations would compromise the efficiency of the design process. CVEs have been studied actively in the research world, but research concentrating on product-centred solutions has been rare. Some exceptions exist, for example, Tuikka (2002) has presented a virtual prototype based solution for synchronous CSCW systems, which provides support for distributed development of personal electronic products.

If the goal is to implement virtual design space as a part of a selected design tool platform, it is advisable to pay attention also to the engineering management and product life cycle management (PLM) systems used in the area of the manufacturing industry and mechanical engineering. These systems are good examples of strong tool platforms capable of embedding the development process guidance and control into their features, thus facilitating the implementation of processes such as virtual design. These systems can, for example, integrate development tools, a PDM system and collaboration tools together and provide for a fluent transformation of design data from one task to another. For example, mechanical design data from a 3D CAD tool can be used directly in setting the assembly line parameters and updating component lists and their ordering in the PDM database. These integrative design systems are usually provided only by large tool vendors that have a strong history and wide product portfolio for specific design or industry areas. For example, Dassault Systems (http://www.3ds.com) offers PLM and engineering management systems for many industry areas with trademarks such as Catia and Enovia. In the software development area, there are similar strong platforms such as Rational Rose from IBM (http://www-306.ibm.com/software/rational/). However, as stated earlier, these systems again clearly demonstrate the problem of diverse design worlds that cannot co-operate naturally. Additionally, even the extensive Rational Rose platform cannot offer tool support for some design areas specific to personal electronics. For example, it does not support rapid product and UI concepting or usability testing.

The second tool category that could have potential in implementing a tool mastered virtual design space is the man-machine interface (MMI) design tools, which have their origin predominantly in embedded system design. These tools are promising as they recognize electronic products as heterogeneous products integrating mechanics, electronics, software and UI design. Some of the tools also include modules for usability testing, and collaboration support via the Web. Many of the tools in this area are focusing on products in which MMI or UI is a part of a larger entity. For example, companies such as Altia Inc. (http://www.altia.com) and eNGENUITY Technologies Inc. (http://www.engenuitytech.com, former Virtual Prototypes Inc.) have a strong background and references in the automotive, aerospace and defence industries. There are also companies that have a focus on consumer and personal electronic products, such as Espial Group Inc. (http://www.espial.com) and e-SIM Inc. (http://www.e-sim.com, former Emultek). e-SIM is an interesting company in the sense that it has been actively promoting the use of virtual prototypes in different product life cycle phases, and as such supported many of the ideas presented in virtual design. The developers of this tool understood the role of prototypes besides implementation, and supported, for example, product marketing with web-based simulation through their LiveManuals (http://www.livemanuals.com) service[22]. In 2002, however, e-SIM changed its strategy away from a broader virtual prototyping approach and focused on the mobile handset industry by offering an end-to-end Mobile Handset MMI Solution for platform vendors, handset manufacturers and operators. In general, the problem with MMI tools is that they are limited to GUI or MMI design, and other tasks and technologies are not recognized or supported. For example, the use of simulations in phases before implementation, e.g. concepting and usability testing, is not supported. Additionally, MMI tools are often implementation technology specific, i.e., they only support a specific programming language and target hardware platform.

[22] The LiveManuals service no longer run as a separate web-page.

6.3 VTT's VRP Environment

The VRP Environment is a research environment developed at VTT for studying, developing and demonstrating VRP technology for small-sized electronic and telecommunication products. The environment was developed within a strategic Virtual Reality Prototyping (VRP) project at VTT, and used in several other research project between 1996 and 2001. The main inventor, architect and realizer of the system was Mr. Marko Salmela. The environment has been described in several publications, e.g., in (Salmela and Pulli 1997, Salmela and Anttila 1997, Kerttula et al. 1997c).

6.3.1 System architecture and components

The VRP Environment consisted of several controlling computers for human-computer interaction, hardware and software simulations. As the goal was to study especially how VR technology could facilitate and provide means for virtual prototyping, a major part of the user interaction was implemented with VR devices. The user, either a developer, a customer or a member of any other interest group, interacted with the virtual prototype through various input and output interfaces (Figure 11). A conventional monitor display with a CrystalEyes stereoscopic system (http://www.reald.com/scientific/) was used to provide visual 3D-feedback. For auditory feedback, stereo headphones and loudspeakers were used. The haptic interface, PHANToM from SensAble Technologies Inc. (http://www.sensable.com) was used in providing the user the ability to feel the structure and mechanical functionality of the virtual prototype. Also the Polhemus FASTRAK tracking device (http://www.polhemus.com) was introduced for tracking the user's hand movements and the rapid physical prototypes kept in the hands.

Figure 11. Main interface devices of VRP Environment.

In the VRP Environment, visual and auditory rendering was performed on an SGI Indigo 2 workstation, and haptic rendering on a Pentium PC. The simulation of logical behaviour and functionality was executed by a VRP Engine, a simulation engine running on a Java-enabled host platform on a separate Pentium PC. A specific VRP Communication System was used for the communication between the different computers. The integration and execution of the parts of the VRP Environment were controlled by the VRP Engine according to the virtual prototype logical model. This model defined the behaviour and functionality of the prototyped target product. The system had an open-architecture and additional external simulation environments, and optional hardware and software simulations could be attached to it through a well-defined communication interface module. One configuration of the VRP Environment is illustrated later in Figure 16, which describes the prototyping arrangements for the ComPen design case.

6.3.2 Component model of the prototype

The virtual prototypes designed and executed in the VRP Environment consisted of the following component models:

- *Integrated visual and haptic rendering models* were used to produce stereoscopic images of a virtual prototype, and tactile and force feedback from the simulated user-to-product interactions. These simulation models were based on the object-oriented OpenInventor 3D toolkit.
- *Optional external hardware and software simulation models* simulated selected components and subsystems of the product, such as embedded hardware and software subsystems.
- *A 3D auditory simulation model* was used for spatial 3D auditory simulation and rendering of the product's sound characteristics.
- *A virtual prototype logical model* was used for specifying a selected part of the behaviour and functionality of the product, and as an integrative simulation model arranging the communication between the other simulator and rendering models.

6.3.3 Process of creating virtual prototypes

The creation and building of a virtual prototype in the VRP Environment consists of the following generic steps:

1. Define the geometry of the virtual prototype, either by converting it from a 3D CAD model, or by modelling it from scratch using an appropriate 3D modelling tool. If necessary, perform optimization of geometry.
2. Fine-tune the surface properties of the geometric model by defining the colours, materials and texture. Again, perform optimization, if necessary.
3. Add the OpenInventor nodes needed for kinematics simulations, user interactions and inter-simulator communication to the scene graph representation of the geometric model. Adjust the parameters of the added nodes.

4. Generate a haptic simulation model from the geometric model and define the tactile and force feedback simulation parameters for it.
5. Create a simulatable, virtual prototype logical model by building it from the existing or new logical model components, called virtual components.
6. Create auditory simulation/rendering models.
7. Integrate the other appropriate simulation models, such as the hardware or software simulation models, to the logical model.

Connect all of the above mentioned component models to the logical object model.

6.4 Cybelius Maestro

Cybelius Software has been producing tools and solutions for virtual prototyping since 1998. The initial business idea of the company was to utilize virtual prototypes in web commerce to provide potential customers with more realistic product experiences. At the beginning of 2001, Cybelius Software took a serious step towards wider support of virtual design by launching the Cybelius Maestro product. Maestro introduced a tool platform for the simulation, prototyping and modelling of electronic and telecommunication products. Maestro was built on the basis of the results coming from the research co-operation between Nokia, VTT and Cybelius Software. The origin of Maestro is in the solutions created in the VTT's VRP Environment[23].

An introduction to Maestro is given in the next chapters. The focus here is on describing the Maestro modelling process. The Maestro approach as a virtual design enabler is discussed later on in Chapter 7.3. General information on Cybelius Maestro is available from http://www.cybelius.com. The Maestro design approach and methodology are described in detail in (Cybelius Software 2005a). When studying Maestro material it is noteworthy that the term simulation is used synonymously with the term prototyping in the material.

---

[23] The main architect of both VTT's VRP Environment and Cybelius Maestro is Mr. Marko Salmela, who currently works at Cybelius Software/CCC Group.

6.4.1 The focus of Maestro

Maestro's introduction text

> "*Cybelius Maestro implements a new kind of design environment, targeted for original product manufacturing companies, product development laboratories and business partners designing product sub-systems, components, software applications and services. Maestro is intended for the design of demanding, new electronics and telecommunications products containing software components and a UI controlled by software. Software features play a key role in these products, but very often they also incorporate several other design disciplines and expertise areas. Typical target products consist of mobile phones including their applications and services, PDA devices, wrist-computers, and navigation, entertainment, audio and multimedia devices for personal, home and car use.*"

taken directly from the document 'Cybelius Maestro – Introduction and Methodology Guide, v1.0' (Cybelius Software 2005a) clearly shows that the aim of Maestro is, especially, to support the virtual design of personal electronic products.

6.4.2 Product family

The Cybelius Maestro product family consists of the following four products:

- *Cybelius Maestro Builder* is a tool environment for the development of functional simulation models and their components. It supports the process of creating simulation models and simulation components, generating code for specific target platforms, and executing generated code in the selected Java environments.
- *Cybelius Maestro Simulator* is a run-time environment for executing Maestro simulation models. Maestro Simulator runs both local stand-alone simulations and distributed simulations over a network.

- *Cybelius Maestro Server* helps in managing the distributed development of Maestro projects. It provides collaborative features and version control support for Maestro Builder users sharing their projects over a local area network (LAN), intranet and extranet.
- *Cybelius Maestro Software Development Toolkit (SDK)* supports the development of new Maestro extensions, i.e. plug-in tools, and Maestro modelling and simulation components. It is a collection of technical documentation, template and example code for developers.

In addition to the basic products, Maestro introduces Extension Packages (EP) that extend their support for specific application areas and tasks. EPs can consist of specific component libraries, plug-in tools, templates, wizards and documentation, and it can extend either Builder or Simulator functionalities. An example of public Maestro EPs is the Maestro Localizer EP designed for distributed UI translation process. Several customer-specific EPs have also been developed, e.g., for the Series 60 and Series 40 UI styles for Nokia.

6.4.3 Prototyping framework

Cybelius Maestro introduces a software tool product family that supports the modelling, prototyping and development of personal electronic products (Figure 12). The basic idea of Maestro is to support efficient development of virtual prototypes and their use in different design tasks under a virtual design process framework. In practice, this means that Maestro supports visual modelling of the virtual prototype in the Maestro Builder, generation of a functional prototype from the model, execution of this prototype in Maestro Simulator run-time environment, and extending the prototype execution with task specific support, cf. Maestro EPs.

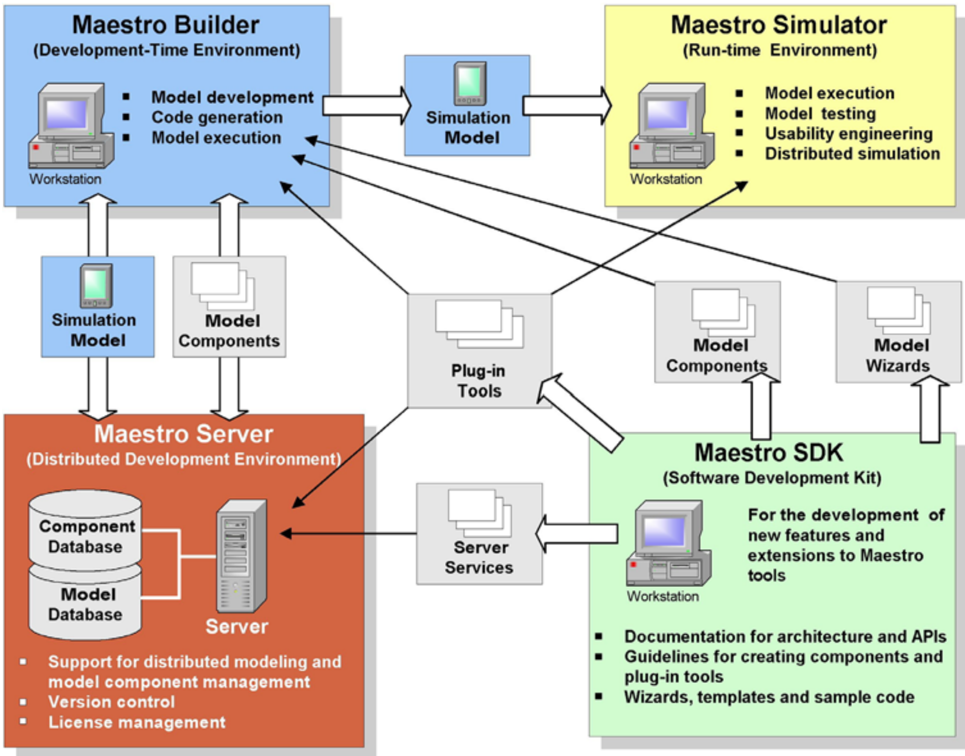


Figure 12. An overview of the Maestro product family. The products and their interconnections.

An overview of the Maestro prototyping approach is given in Figure 13. The Maestro simulation model presented in the middle of the illustration consists of the sub-models that are modelled with appropriate editors. Sub-models can incorporate or interface with external components, resources and tools, which means that real objects can also be used as a part of a prototype model and its execution. External objects can include:

- exact graphic resources, such as icons and fonts
- software components, such as a component implementing predictive text input
- external tools integrating simulation, for example, with a mobile network simulator
- external devices providing access, for example, to a camera used as a part of the prototype.

The integration of external objects to the Maestro simulation model is supported with a Maestro Connectivity Framework that offers documented and open interfaces for integration. During the model execution, objects are allocated for the simulation engine via the Maestro Simulation Bus.

After a prototype has been generated from the simulation model, it can be executed in the Maestro Simulator. The UI parts of the prototype are displayed either within the run-time environment running on a PC, or in a separate hardware mock-up connected to the run-time environment wirelessly or wired. The customized code generation module of Maestro also enables target specific code generation, which means that the simulation model or its parts can also be executed in a real target product. Code generation EP is often offered as a part of a target specific EP, including project templates, modelling components, graphics resources, application models and code generation plug-in(s) for a selected target platform or product.

Component-based modelling is one of the base elements of the Maestro approach. Components do not only provide functions and services internally for the simulation model, but they help in integrating the simulation model into the real world. Components providing interfaces to external software modules and the existing products enable the construction of heterogeneous simulations mixing real and simulated entities in an executable simulation experience. Components also provide a possibility to generate target-specific code for different platforms, as the information of the target platform can be integrated into the simulation components used in building the simulation and into the components describing the simulation architecture and target environment.

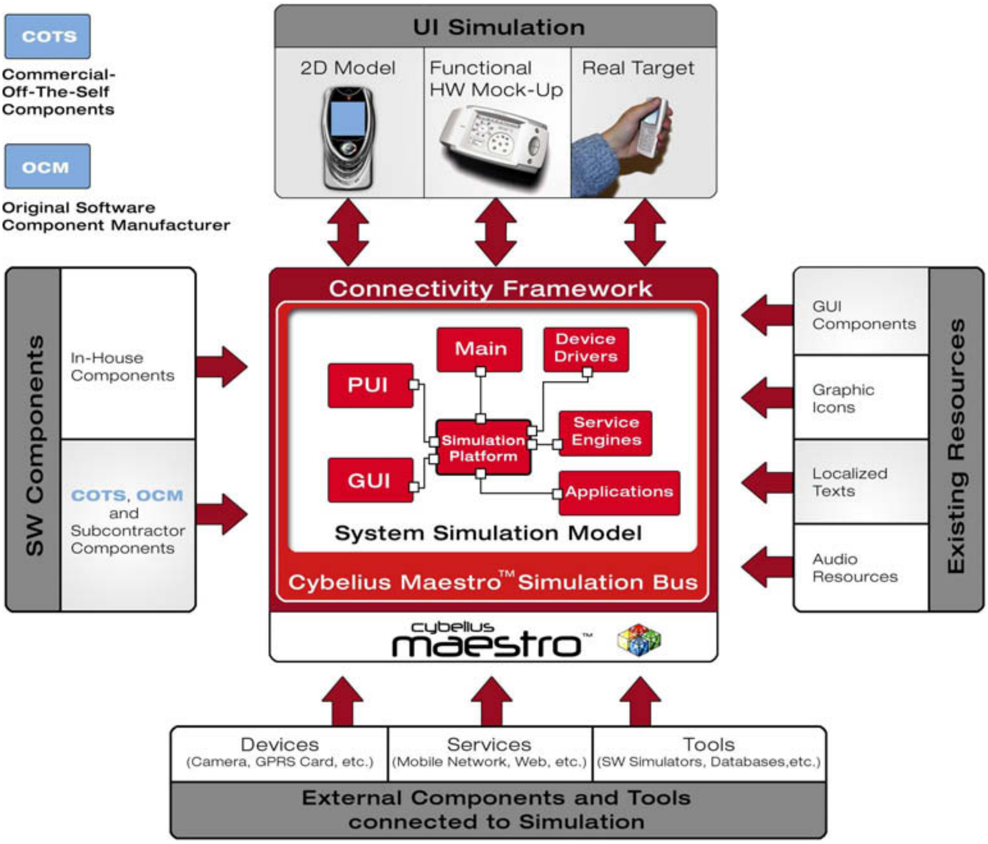


Figure 13. Cybelius Maestro prototyping framework.

6.4.4 Simulation model structure

The Maestro simulation model defining the virtual prototype consists of several sub-models, which are created and modified with the corresponding editor. The model structure can be generalized to include the following main elements:

- *Physical User Interface (PUI)* describes the physical form of the device: the industrial design and mechanical casing with its assembled external parts. This model is important as it gives an immediate impression of the product and provides a naturally understood symbol for the target of the entire project. The model can be represented by visual computer generated images, and later also by physical mock-ups of real product prototypes, and during the final design phases even by the product itself. As an extension, the design environment provides voice input and output channels, and their controls. From the virtual design point of view, the PUI model provides an important link from the software-centric design world to mechanical and industrial design.

- *Graphical User Interface (GUI)* simulates the graphical user interface elements seen on the display(s) of the product. It presents the texts, icons, images, et cetera, for the user exactly as they would appear in a real product. From a design process perspective, the GUI model has a strong link to the process phases in which decisions on general product style, GUI style and UI style are made. Naturally, it also directly links to the GUI design and implementation task.
- *Software Model* implements the overall behaviour of the product and its software architecture. Behaviour is modelled by using a state machine formalism based on the Unified Modelling Language (UML), or directly by code in the software components. The software architecture is described by the System Software Structure Model. This model provides a means to adjust the simulation model for specific target environments. It can, for example, define the hardware interfaces and operating system services available in the target platform. Software Model refers directly to the software design and implementation phases in product development. If target platform architecture is taken into account in the System Software Structure Model, for example, by providing the needed Application Programming Interface (API) methods to state machine modelling, the Maestro behaviour model can directly support the software design and implementation. State machine printouts can be used, for example, in software specifications, and generated code or parts of it can be directly made use of in the software coding.
- *Resource Model* defines text and graphics resources for GUI elements and the PUI model. Resources, such as text, icons, lists and animations, are referred to by symbolic names that allow product localization support for different languages. The Resource Model directly supports the localization and translation of the product UI. It allows the design and testing of different text translations parallel to UI design in their right use context.

- *Applications* are a specific type of sub-models that usually have their own Behaviour, GUI and Resource models. Applications can be exported and imported between the Maestro simulation model projects. Application support pushes Maestro more deeply towards the current mobile architectures, in which applications play an important role. In the wider sense, applications resemble the collaborative nature of personal electronic business. Today, many of the product parts are made by partners and subcontractors. Application export and import is a way to support this distributed development by allowing the integration of design results and ideas together already in the early prototyping.

The appearance of the modelled product is created through traditional visualization techniques. Built-in tool resources include ready-made product templates for a fast start-up, in which users can further supplement them with their own appearance images, skins, using off-the-shelf graphical drawing tools or imported pictures. In product simulation these images are connected with the executable models of the logical product behaviour.

Maestro provides pre-defined and pre-programmed simulation components and modelling templates for the users. Basic component libraries and templates can be extended further with custom-made items by programmers. This component-based approach makes it possible that users without programming skills are able to create even complex simulations.

6.4.5 Generating and executing the prototype

Whenever a Maestro user has defined and finished a product model, the model can be compiled into executable Java code. By default Maestro Builder supports code generation for the Java 2 Platform, Standard Edition (J2SE) and J2SE Applets. The model can then be executed using the Maestro Simulator or Maestro Viewer. Any standard type personal computer running Java 2 can be used to execute the default Maestro simulations. The computer's display and audio properties are employed to create the impression of what the product looks like and how it feels to use it.

Maestro Simulator offers a run-time environment that can be extended with Maestro run-time plug-in tools. The run-time tools offer additional support for performing different tasks, such as usability testing, with simulation. The Simulator also supports distributed and connected simulations running synchronously in several Simulators. The Maestro Viewer is a utility for the lightweight execution and distribution of generated simulation models. The viewer can be packed into the same JAR file with the generated model to provide a stand-alone, executable JAR file. This package can be distributed, for example, via the Web or e-mail for product promotion, evaluation, training and customer service purposes.

Maestro's default code generation plug-in can be substituted or extended with custom made plug-ins, either by the user company or in co-operation with Cybelius. This customized code generation enables running generated software modules on a specific software platform or operating system. Maestro's code generation framework is totally component-based, which means that code generation is customizable not only to Java platforms, but also to other possible targets. Customized code generation is implemented with customized, platform-specific EPs that usually include target-specific code generation components, a modelling framework and a code generation plug-in. Cybelius Software has already developed code generation EPs for different C/C++ target platforms. One of them has also been released publicly and explained in a press release published on February 11th, 2005 (Cybelius Software 2005b).

6.4.6 Base technologies and architecture

The leading design principles of Maestro are openness and flexibility. Maestro has been built on an open and expandable tool platform and architecture, which is based on Java and XML technologies. The Maestro tool architecture describes the structure and architecture of Maestro tools. All Maestro products are based on the same component-based plug-in architecture, which is built on Java technology. This architecture is also called common Cybelius Product Platform. The main elements of the platform are:

- *A Plug-in Tool Bus,* which connects Maestro plug-in tools to the platform and its core services.

- *Core Services*, which are internal platform components offering basic services for the product platform.
- *Plug-in tools,* which are the actual tools provided by a Maestro product for the users. For example, in Maestro Builder, editors such as the PUI Editor and the GUI Editor are plug-in tools.

6.4.7 Support for customization

Maestro is based on a unified software architecture with well-defined module interfaces. Internal Maestro modules conform to the Java software component concept. For this reason, the product and its modules are platform-independent. Maestro users who are familiar with Java may utilize and extend tool resources efficiently. Existing products and product components can be combined to set up new simulation configurations and to create fully functional system simulations. Furthermore, various system configurations can be tested with different peripherals, accessories and interface components. Even though Maestro itself uses Java implementation, also non-Java components and tools can be integrated with it through appropriate interfaces.

Maestro tools, components and working environments can be customized for specific needs. Maestro can be supplemented, even by the users themselves, with new plug-in modules, to extend its functionality and application possibilities. Extending and customizing Maestro can take place according to the following three approaches:

- Developing company and product specific components and templates. For example, the software and user interface features of a product family can be implemented as components so that they are easier to apply in new product versions. Company-specific resources, such as fonts and icons can be developed and shared.
- Developing interface components and plug-ins that link Maestro to other systems and tools used in the company. For example, a Maestro user can fetch localized product resources from the company's existing resource database. In addition, interface components for supporting selected hardware

platforms and components can be built so that real hardware can be used in simulations, either as a background resource or as an execution platform.

- Extending the Maestro platform with new plug-in tools that support company-specific tasks. For example, a special code generation plug-in tool for customer-specific target platforms can be developed. With this extension, a customer can generate native code for their products directly from Maestro models. With this arrangement, the use of Maestro is expanded naturally from the simulation to the implementation area.

A typical way to realize the above customization is to collect all of the needed extensions for a customer-specific or for an application-specific Maestro EP.

7. Design studies

The potential of virtual design in the development of personal electronic products is evaluated by three design studies. The studies consist of 2 public cases and a summative case describing the development of mobile phone UI software with virtual design. The first public case was implemented between 1996 and 1998, and it studies the possibilities of VRP in the concepting of a new personal electronic product idea called ComPen. The second public case, Cybeo ONE, was introduced in 2002 at the CeBIT exhibition, in Hanover, Germany. It was implemented with the Cybelius Maestro tool and was presented as a part of the Cybelius Software demonstration. The summative case of mobile phone UI software design summarizes the experiences of applying the Cybelius Maestro tool in real customer projects between 2001 and 2005. The cases demonstrate the results of virtual design both in specific design tasks and as a wider product development process framework.

7.1 ComPen – a study in VRP

ComPen, a pen-like wireless interface device for a mobile phone, was a public pilot case developed in a national research project on Virtual Prototyping Services for the Electronics and Telecommunications Industries (VIRPI) during 1996–1998. The idea was to study a product concept of a mobile phone accessory that could be used:

- to wirelessly access the UI of the phone to which it was paired, i.e. the master phone;
- to control the functions offered by the master phone; and
- to make GSM phone calls via the master phone.

On the master phone side, a real ComPen implementation would have required an accessory module to interface ComPen to the phone's internal bus via an appropriate connector.

Figure 14. ComPen designs – a pen-like wireless interface device for a cellular phone (ComPen model courtesy of Metsävainio Design Oy, Finland).

The concept of ComPen was invented by professor Petri Pulli at VTT, and the practical work was done using VTT's VRP Environment. Virtual prototyping and VRP were applied in searching for appropriate industrial designs (Figure 14), and for an UI design including UI logic and features. The usability of the ComPen was also tested with the VRP approach. Besides concepting there was a subproject that studied the real electronic implementation possibilities for the device. Here it is noteworthy that the work was performed mainly between 1996 and 1997, when there were no commercial short-range wireless technologies available, such as Bluetooth[24] or WLAN[25] (Wireless Local Area Network). In

[24] Ericsson Mobile Communications initiated a study on Bluetooth in 1994. Bluetooth Special Interest Group was initiated by Ericsson, IBM, Intel, Nokia and Toshiba in September 1998. The first Bluetooth products came on the market in 2000.

[25] The Institute of Electrical and Electronic Engineers (IEEE) introduced specifications for 802.11 in 1997. It outlined the protocols for WLAN connections with LAN-equivalent speed and security. More popularly known as Wi-Fi (wireless fidelity), 802.11 provided wireless transmission rates of 11Mbps.

that sense we were prototyping a future product concept that was not realistic to implement with the technology available at that time, but which was expected to be feasible in the future, if technology would advance at a predicted rate of progress. ComPen was also interesting because of its business potential. If it had have been possible to produce a wide variation of customized ComPen models, i.e. different design and features for different user groups, at a relatively low price, the product could have successfully entered mass markets. It would have had potential as a replacement of your phone models, i.e., instead of updating your expensive mobile phone you could only have selected a new low-cost ComPen model.

7.1.1 Development work

The development of ComPen was started with an industrial design by using 3D CAD and CAID tools. Design variations were produced by Metsävainio Design Oy, a design agency that participated in the VIRPI project. Already in this phase different UI choices were studied with combinations of different menu structures, display segments and input elements, such as slide buttons and rocker switches. Some of design variations are presented in Figure 14. A five-button UI was selected for further implementation (Figure 15). Next, the functionality of the ComPen was defined and the general architectural structure of the device was decided and modelled with the Real-Time Object-Oriented Modelling (ROOM) method (Selic et al. 1994) using VTT's VRP Environment. The key issue in the UI design was the ease of use, and so the selection of the available functions was kept small by concentrating only on the most essential features. Functionality i.e. the virtual prototype logical model of the ComPen was implemented with Java[26] and transferred to VRP Engine, the simulation engine of the VRP Environment.

---

[26] Here it is important to note the forward looking thinking of Mr. Marko Salmela. He had confidence in the capabilities of Java already in 1996 and he predicted that it would have a major role also in future wireless devices. Sun Microsystems Inc. introduced the K-Virtual Machine for CLDC/MIDP devices at 1999.

Figure 15. A five-button design of ComPen (ComPen model courtesy of Metsävainio Design Oy, Finland).

Parallel to the UI design, the work for integrating a Nokia 2110 GSM mobile phone to the VRP Environment was started. This was done by using Microsoft's Telephone Application Programming Interface[27] (TAPI) and the Nokia Telephone Service Provider[28] (TSP) library. The remote control of the GSM phone via its controlling computer was implemented using TCP/IP (Transmission Control Protocol/Internet Protocol) messages (Nalli 1997). The communication protocol was designed according to the specification of the VRP Environment, and the TAPI client. Here, the TAPI software was controlling the GSM phone through a PC attached to the VRP Environment as an external simulation system (Figure 16). With these arrangements all of the functions of a GSM phone were available through the network for the VRP Environment. TCP/IP remote control messages were integrated into the ComPen logical model. With this addition, ComPen simulation in the VRP Engine had access to all of the menus and menu items of the GSM phone and calls to the GSM network were able to pass through it.

On the visual simulation side, the ComPen 3D model was transformed into a OpenInventor (Wernecke 1994) 3D format and loaded into a SGI workstation. This model was called a visualization model of ComPen. A PHANToM haptic interface device was attached to the simulation (Anttila 1998) in order to provide a complete feeling of a product that does not actually exist in the real world. In practice, this meant that the haptic rendering model of ComPen was generated from the 3D geometric model by setting the tactile and force feedback simulation parameters for each virtual plane in the geometric model. Finally, all of the system simulation parts, i.e. visualization model, haptic model, logical

[27] Windows Telephone Application Programmer's Guide, version 1.0, Microsoft, 1993.

[28] NOKIA telephone Application Programming Interface. Development kit, Version 1.0, Nokia Mobile Phones, Cellular Data, 1995.

model, were connected together and executed under the VRP Engine. The result was that the final simulation of ComPen had actual GSM features, and was able to be interacted via stereoscopic glasses, a mouse and a keyboard, and also through a haptic interface (Figure 11).

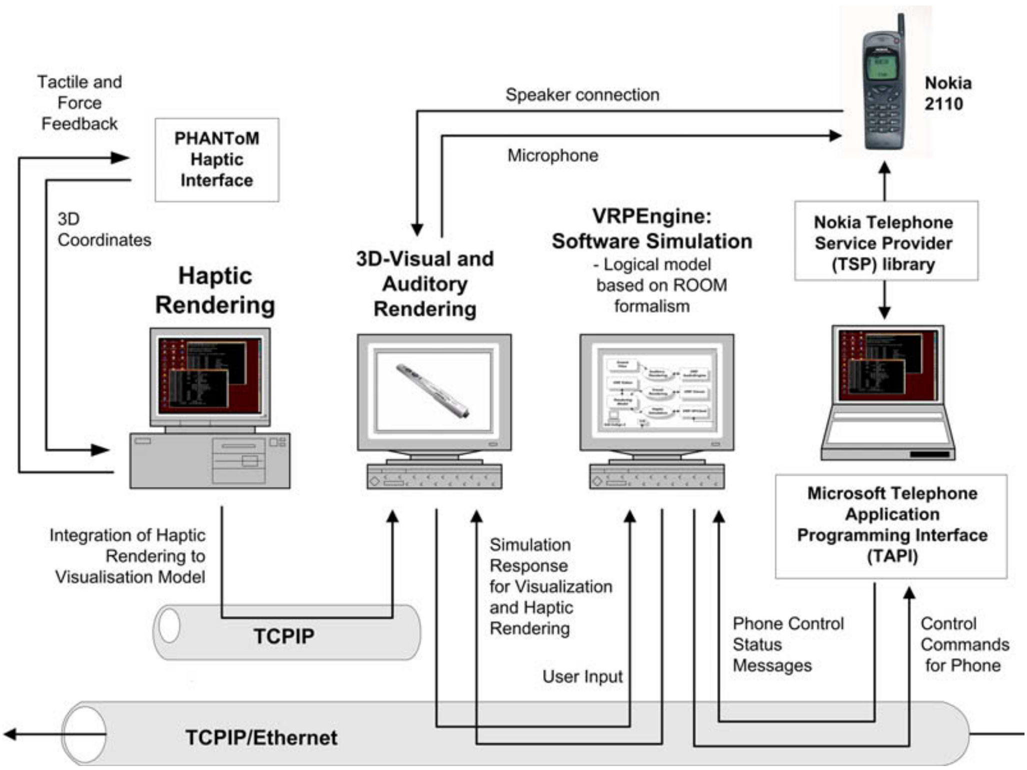


Figure 16. ComPen virtual prototyping arrangements.

7.1.2 Results

The ComPen design case was a successful demonstration of the VRP Environment and VRP approach in general. It enabled an open case without the normal limitations of industrial cases, and as such provided an interesting field of discussion between the researcher and industrial partners. The ComPen realization implemented with the VRP Environment presented many of the main characteristics of virtual prototyping, VRP and virtual design.

From the point of view of a general virtual prototyping approach the case showed that:

- Virtual prototyping enabled the creation of a functional, interactive and visually accurate presentation of ComPen.
- Virtual prototyping enabled the creation and testing of a functional product concept that was not possible to implement with contemporary technology.
- The Virtual prototype was able to integrate different design areas into a common product presentation. ComPen integrated industrial and mechanical design into software design.
- Virtual prototyping supported heterogeneous prototyping by integrating existing hardware or products with simulated product features in order to create a realistic product presentation and user experiences. In the case, a real GSM phone was integrated into the ComPen simulation.

When studying the case from the perspective of VRP, it was found out that:

- VR technology was able to make the user experience immersive even in a desktop simulation.

ComPen was a limited design case covering only a restricted number of design tasks compared to real product development. When the case was implemented, the aim was not to study it from the design process perspective. However, even as such the case proved some important issues that supported the implementation of virtual design:

- Virtual prototype supported the early validation of a new product concept.
- Virtual prototype supported the communication between different design disciplines.
- The same virtual prototype could be reused in different tasks. In the ComPen case, the prototype was used in the validation of design decisions and later in many demonstrations of the technology. In real product development these tasks can be considered equal to the tasks of product development decision making, usability testing and marketing.

It is important to note that ComPen prototypes were also used later on in other tasks of the VIRPI project. Tuomo Tuikka from the University of Oulu used ComPen in his research in CSCW using virtual prototypes (Tuikka 2002), and Battarbee et al. (1999) used the same model when studying the use of 3D virtual prototypes in usability testing over the Internet.

When analyzing the ComPen case some years later, the results of the case can be extended with one important conclusion. Even though the interaction experience of ComPen could be extended with VR devices, it was noticed later that VR interaction was not well adopted by companies developing small hand-held devices. The main reasons for this were:

- VR devices were still expensive compared to normal PC computers and peripherals.
- Set-up and maintenance of the VR devices was demanding.

Virtual reality prototypes and their use was not comparable enough to a normal use situation. For example, the use of the PHANToM haptic device in a ComPen use situation was unnatural, as the user was pressing buttons located in the air, not by way of a mechanical object kept in the hand. Additionally, the users often became confused, and even scared, with the VR devices.

Figure 17. Cybeo ONE design.

7.2 Cybeo ONE – hardware mock-up demonstration

Cybeo ONE is a phone demonstration concept implemented by Cybelius Software Oy (http://www.cybelius.com) at the beginning of 2002. The motivation behind the demonstration was to create a public demonstration that could be used to demonstrate the features of the Cybelius Maestro prototyping tool platform. Cybelius had interesting industrial projects and customer cases, but it was impossible to show any results from them publicly because of confidentiality reasons. Thus there was a clear need for a public demonstration case. Besides simply focusing on software features shown on a computer screen, the goal was to implement a concrete, attractive and eye-catching device demonstration that would have better possibilities to attract some attention at marketing events, such as trade fairs. For this reason, it was decided that the Cybelius Maestro simulation running normally on a PC screen would be extended with a wireless hardware mock-up. The simulation screens from the PC were redirected to the display of the mock-up and the control commands, such as key presses, were received from the mock-up and used in order to control the execution of the simulation.

On a functional level the Cybeo ONE concept phone incorporated a number of possibilities of future smart phones. A large touch screen based design was selected as a starting point in the design (Figure 17). A large full colour display enabled the concepting of new UI design styles. On the application side, the goal was to demonstrate new mobile applications including multi-media messaging, e-mailing, imaging and video. Here it is important to notice that most of the previous services were not supported either by the phone manufacturers or mobile operators at that time. Additionally, Cybeo ONE included a feature to make real phone calls into a GSM network. Cybeo ONE was introduced for the first time at the CeBIT fair 2002 in Hanover, Germany. The design of the demonstrated Cybeo ONE is shown in Figure 17, and the demonstration architecture is illustrated in Figure 18.

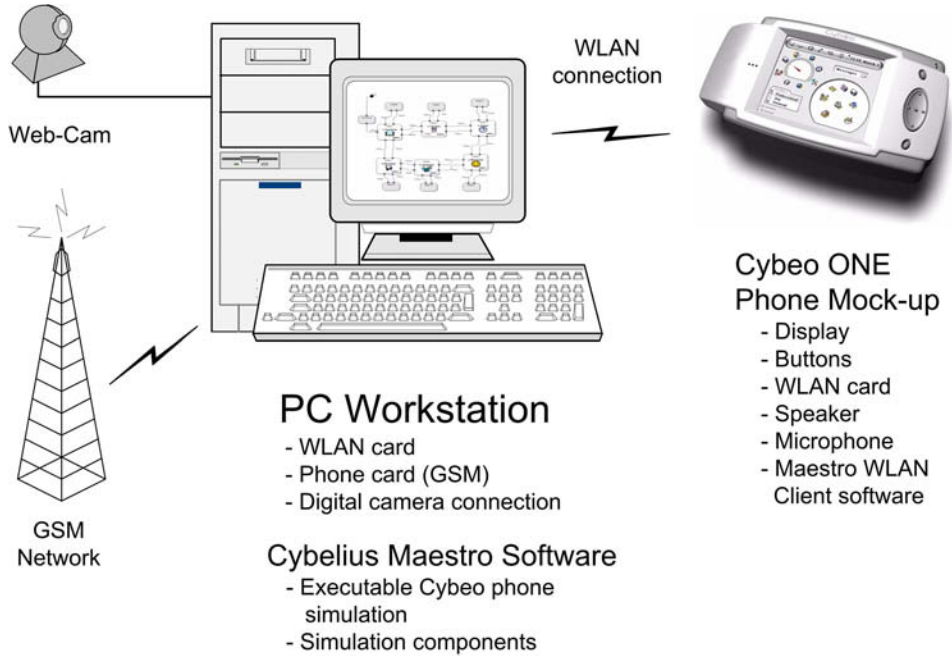


Figure 18. Cybeo ONE demonstration arrangements.

7.2.1 Cybeo ONE development

Cybeo ONE development started in November 2001, and it was carried out as a co-operation project between Cybelius Software, VTT and RPC Group Oy (http://www.rpc.fi). The basic requirements and constraints for the project were set at the project start-up meeting:

- Due to a tight schedule a Compaq iPAQ Pocket PC[29] device was selected as the base for the mock-up electronic implementation. The selected iPAQ model 3850 included a WLAN extension card for wireless communication, and on the PC side WLAN was accessed via a separate WLAN card. The screen size was set to 290 x 240.

[29] Note: at present iPAQ Pocket PC is sold by Hewlett-Packard Development Company, L.P. under HP iPAQs trademark after acquisition of Compaq Computer Corp.

- The device functions would mainly be operated through the touch screen using pen-based interaction.
- The wireless connection should also carry voice and screen images and control messages.
- The iPAQ device would be dismantled and covered with new covers produced with a rapid prototyping technique. The cover should allow access to the main controls of the underlying iPAQ.
- Cybeo ONE simulation should include access to a real GSM network through a GSM card (Nokia D211) and a camera connection enabling imaging applications.

Figure 19. Cybeo ONE UI styles. Youngster and professional styles.

Additionally, the project was divided into the following sub-tasks:

- development of the Cybeo ONE simulation
- development of wireless communication software
- design and manufacturing of mechanical parts for Cybeo ONE covers
- dismantling of iPAQs
- assembly of a pre-series (5 pieces)
- integration testing.

The development of the Cybeo ONE simulation project was started by defining the features for the demonstration. The functions selected for the first demonstration included:

- browsing the menu levels
- making a call:
  - entering a direct number (no real call)
  - choosing a name from the contacts
  - only one predefined contact would be accessible for a real GSM call, others are prohibited in CeBIT
- answering a call
- sending an e-mail, SMS or MMS message:
  - input text into fields
  - select attachments
- viewing a web camera picture:
  - freezing the image
  - sending an image "souvenir" to a given e-mail address.

Parallel to the function specification, the Cybelius graphical and industrial designers started to work with the general Cybeo ONE design. They introduced several UI styles and some industrial designs for the cover. The design given in Figure 17 was selected for implementation. Two UI styles called youngster style and professional were selected for implementation. They are shown in Figure 19. The styles had a different appearance, but they shared the same logic for their operation.

As soon as the features for the demonstration were set, the development of the Maestro simulation project was initiated. The project included the development of GUI screens using graphics from designers, the PUI model, and the behaviour modelling of demonstration features. Additionally, software components for web camera access and a GSM phone card were developed.

The development of WLAN communication software for iPAQ and the Maestro Simulator was started by VTT. The communication software was implemented mainly with Java and selected C-code DLLs. For the iPAQ, a client software was written for establishing the connection, taking care of redirecting screen images coming from the PC to iPAQ's display, forwarding touch screen actions from the iPAQ screen to the PC, and for establishing a two-channel voice connection from the iPAQ microphone to the PC audio out connection and from the PC audio into the iPAQ loudspeaker. The PC audio in/out connections were further used to connect an analogue voice to a GSM phone card.

After the Cybeo ONE cover design had been selected at Cybelius, the 3D images were sent to the RPC Group for rapid prototyping. After the necessary small modifications to the 3D files had been made, a master prototype was created using stereolitography (SLA) technology. The reassembly of the dismantled iPAQ to the master prototype was tested and after successful modifications a series of 5 covers were created with silicon moulds and polyurethane casting. The covers were finished and painted, after which they were assembled with iPAQs. The integration testing was finalized just before the CeBIT fair in March 2002.

7.2.2 Results

Cybelius had two Cybeo ONE devices at the exhibition booth at CeBIT. They functioned well, both technically and as demonstrations. The user of the device was able to study different UI designs of the youngster and professional styles, switch rapidly between the styles, see live video stream coming from a web camera, freeze the image and send e-mail greetings with an image to freely selectable e-mail address, and make a GSM phone call via the contacts menu or by direct dialling. During the exhibition, the use of a real GSM call was disabled, because it was noticed that the audio properties in the mock-up design were not good enough. A simulated call connection and answering voice were used instead.

After its first appearance in CeBIT, Cybeo ONE has been successfully used as a demonstration device at many sales and marketing events between 2002 and 2004. It has been also used with other Maestro simulation projects, such as the one using a Nokia Series 60 UI style. New implementations of the Maestro mock-up extensions have been done based on Cybeo ONE experiences. At the ITU Telecom World 2003 Exhibition, in Geneva, Switzerland, Cybelius demonstrated a similar wireless mock-up idea with a Jappla hardware platform development by the Nokia Research Center. Further similar solutions using Bluetooth wireless technology instead of WLAN have also been researched.

Cybeo ONE was a demonstration of the Cybelius Maestro simulation, modelling and development tool. It introduced not only the design result of Maestro modelling, but also a design project modelled with Maestro. The concept proved some of the important features of the Cybelius Maestro tool platform:

- By using Maestro it is possible to build multi-disciplinary virtual prototypes integrating different design areas. In the Cybeo ONE project, the integration included software, industrial and mechanical design.
- Maestro enables the building of a concrete product concept as a virtual prototype that is functional, interactive and visually accurate. It also enables a fast modification and update of the concept by supporting the editing of the necessary sub-models of the prototype model.
- Maestro enables the demonstration and validation of product concepts that are not possible to implement with contemporary technology. In the Cybeo ONE demonstration, not only a new device was prototyped, but also new mobile services (e-mail and MMS), and applications (imaging and video streaming) were demonstrated.
- Maestro supports heterogeneous prototyping. In the Cybeo ONE project, a real digital camera and a GSM phone card were integrated into the demonstration as real product components in order to enhance the prototyping experience.

Cybeo ONE showed that it was possible to build a hardware mock-up platform that could also incorporate software functionality into usually applied dummy mechanical prototypes created with rapid prototyping technologies. Additionally, in the Maestro-based mock-up solution, the software functionality was not implemented by hard-coding software to mock-up electronics, but with a Maestro simulation based approach enabling quick alteration of the software features in the mock-up.

In addition to the demonstration use of Cybeo ONE, the mock-up implementation created important components that were later used by Cybelius in real customer projects. The experience from Cybeo ONE, its successors and customer cases using Maestro-based mock-ups indicates that functional mock-ups offer a good way of demonstrating new product concepts. It is clear that with software-focused products the mock-up must also be able to show software functionality. With this property the mock-ups are of benefit especially in the business of personal electronic products. They can be utilized in early concepting with test user groups, in later phases of usability testing, product marketing, e.g. in exhibitions, and as a general means of support when important product decisions are made. Additionally, it has been noticed that when mock-ups are used in the early concepting and demonstration phases, it is important that the changes to the software-based mock-up functions can be performed quickly. Thus even real-time modification of the software (UI and applications) should be supported. In this respect, developing software prototypes into an existing product with a native programming language is often too slow and laborious for these design phases.

When analyzing the ComPen and Cybeo ONE design cases from the perspective of virtual design research, they show an important development in the needs of user interaction simulation. ComPen showed that VR technologies did not succeed in companies as an everyday prototyping technology. Cybeo ONE proved that functional mock-ups mimicking a more realistic use situation of personal electronics has more potential in practical use. However, the progress has not stopped in this track. Based on the latest findings in Maestro customer companies, the needs in UI simulation are shifting towards the requirements to simulate UI styles, and applications and services running quickly on the screens quickly of existing devices. The possibilities of virtual design and the Maestro approach to support this requirement are discussed within the next summative design case.

7.3 Development of mobile phone UI with virtual design

The last design study introduces a summative evaluation of virtual design in the development of mobile phones, especially regarding its UI software. The study summarizes the experience from several industrial cases (ME, MA, MJ, MU, OE, DE and DJ in Table 1) that have been participated in at Cybelius Software between 2000 and 2005. The companies include global mobile phone manufacturers, mobile operators and OEM/ODM companies from Europe, Japan, Asia and the USA. A summative approach is selected here so that the publication of exact company-specific information can be avoided. Even though the product described here is a mobile phone, the material and findings also apply to the UI development of many other personal electronic products, such as PDAs, wrist computers, and the like. It is also worth noting that this is a summative case presentation, which does not aim to be a complete or detailed description of real mobile phone UI development process.

A mobile phone is a good example of a personal electronic product on the market. When looking at it from the technical point of view it really stands for a heterogeneous product comprising a wide scope of expertise areas such as industrial design, mechanical design, hardware design, RF (Radio Frequency) design, software development and usability engineering. It also represents a software-intensive product that is targeted to consumer markets. These facts suggest that fluent internal communication in the development process, emphasis of software design practices and understanding the role of good design and usability are fundamental elements for a successful mobile phone business.

The mobile phone business in general sets strict requirements for product development. The mobile phone is already an established and stable product, and the mobile phone industry has shifted from the pioneering phase into a normal competitive situation. Mobile phone companies are competing with their specific strategies, which may include volume prices, new features, phones for different market segments, fashion and brand values, and so on. The common denominator for this situation is that companies have to continuously introduce new models to the market. However, from the technical point of view, most of the new models are evolution models of current phones, i.e. re-designed products with new attractive features.

The requirements presented above are still expanding with the nuances coming from global business, i.e. distributed product development, localized products, global markets and partners set their own specific requirements for mobile phone design. Mobile phone design often takes place in a modern, global organization, which operates at different sites, and controls and manages its functions efficiently. The product process is distributed into different locations, which gives rise to the following special needs to be satisfied:

- Geographically distributed designers and teams must have the means to share their work efficiently between each other.

- Reviews and verification of design results need to be managed in a distributed environment. This means that design results should be easily distributed and discussed over a corporate communication network.
- Control of travelling. This goal is associated with the control of the total costs in company projects. The amount of travelling is also connected with work satisfaction and with the safety of the employees.

7.3.1 UI software development

The case focuses on the UI software development of a new mobile phone model. UI software refers to the higher-level software parts implementing:

- the graphical design of UI, including icons, fonts, animations and screen layouts;
- the logical structure and main functions of UI and their associations with controlling keys; and
- selected applications running in the UI.

The UI software is running on top of a mobile phone system software that consists of an operating system and middleware. The system software can be a proprietary design specific only to a given company and hardware platform, or an open software platform enabling integration into several hardware platforms and being used by many phone manufacturers. Open software platforms promote active 3rd party developer communities that offer services for platform customization and for the development of platform specific applications. Examples of open mobile phone platforms are Symbian OS (http://www.symbian.com), Microsoft Windows Mobile (http://www.microsoft.com/windowsmobile), and Qualcomm Brew (http://brew.qualcomm.com/brew).

The development of UI software includes the normal phases of design, implementation and testing for the graphical UI components, UI functionalities and applications. The software development practices and tools vary a lot depending on the target hardware platform. Usually, the open mobile phone platforms have a wider selection of developer tools, such as platform specific SDKs operating together with major IDEs from Microsoft, Borland and

Metrowerks. These tools can hide lower-level design complexity and raise productivity with component reuse and a wide selection of available APIs. Another group of tools consists of proprietary embedded software development tools that are targeted to specific hardware platforms. Typically, they offer low-level libraries, editors, cross-compilers and debugging tools, and require more low-level programming expertise. The graphical design for UI is also performed in various ways. With advanced development environments, UI graphics can be designed by using predefined GUI components providing basic widgets, such as buttons, icons, and lists that are extended with customized graphics imported from graphical design tools. In many proprietary platforms the selection of GUI components is restricted and there are no tools supporting the visual design of UI.

UI design has a significant role in mobile phone development. It has a direct impact on the success of the mobile phone UI, which, indeed, has become one of the strategic competitive factors in the mobile phone business. The success of the UI contributes directly to the desirability and usability of the phone model. Additionally, the UI is the main tool for manufacturers for doing phone segmentation for different user groups (Kiljander 2004). On the other hand, as a design task or process, the UI interfaces with and links to many other design areas and company functions that together build the success of the business (Figure 20). The other tasks or processes linked to the UI design process or its results are:

- *Concept design:* Simulations, visualizations and prototypes are needed when new mobile phone variations are sought. Especially software functionalities are difficult to experience without any practical demonstrations illustrating the operations performed with software. This means that either real software or their simulations are also needed on the UI level in order to enable a better validation of design choices.
- *Industrial design:* The integration of industrial design and UI software design in the early design phases is becoming increasingly critical. The size of the screens is growing and it is important to ensure that the industrial design focusing on mechanical parts and the UI software design implementing the outlook of screens suit each other.

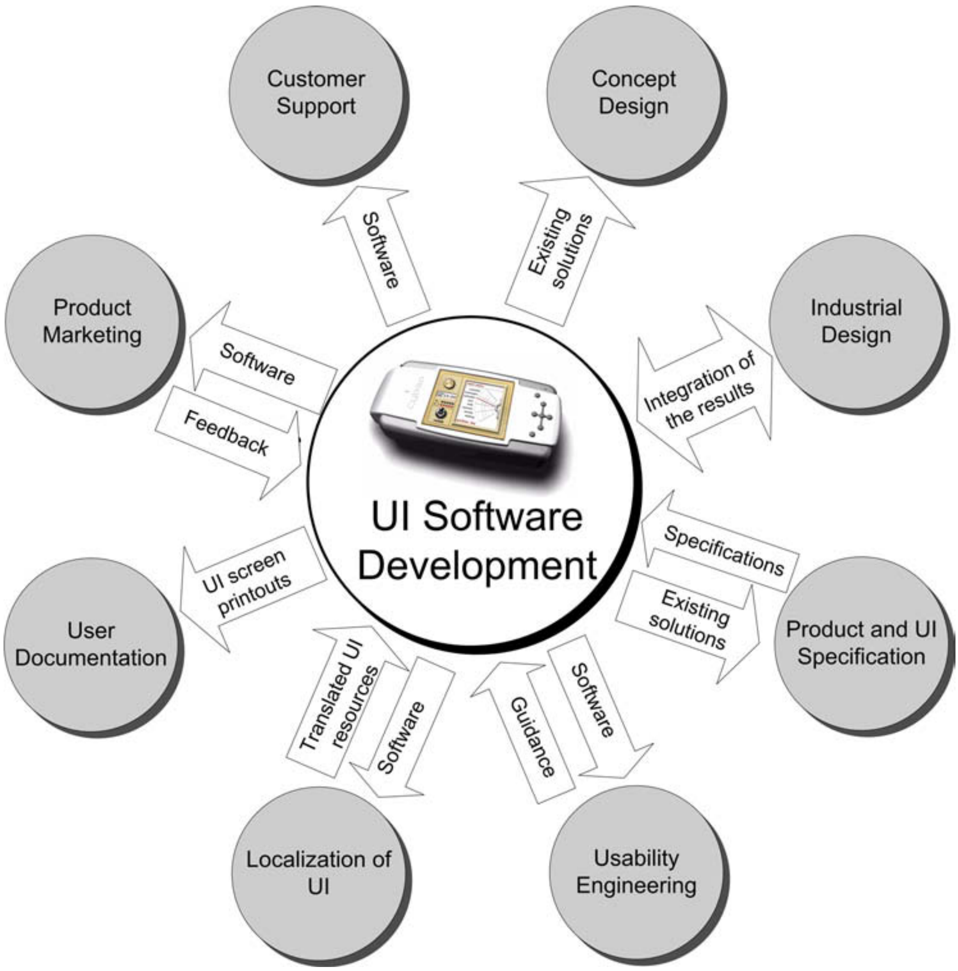


Figure 20. Central role of UI design in the mobile phone development process.

- *Product and UI specification:* After the decision to develop a new product has been made, the specification is started. The specification also defines exact UI behaviour and the functions of the UI specifications.
- *Usability engineering:* High quality and easy-to-use UI and UI software are the major contributors for good usability in mobile phones. Usually, most of the effort in usability engineering is placed on UI related issues.
- *Localization of UI:* Turning UI functions and elements into several languages is a laborious task. This work requires a good understanding of translated functions, and it can usually be performed only after the UI design has been finished.

- *User documentation:* Mobile phones user manuals predominantly focus on describing the functions of the UI. Most of the images used in manuals are based on direct utilization of UI screen printouts in specific operations.
- *Product marketing and sales:* The advantages and features of a new mobile phone model are strongly related to UI and UI software. This means that the marketing people must be able to show and demonstrate the software of the phone, which is not an easy task with traditional marketing means.
- *Customer support:* Here the challenge is equal to the one in marketing. Describing the software functionality in phone support orally or on the Web with the traditional means of images and visualizations is not an easy task.

The development of UI software itself includes the normal phases of design, implementation and testing for the graphical UI components, UI logic and functionalities and applications.

7.3.2 Current practices and problem areas

Reaching a fluent and effective co-operation between the UI design and other operations is a great challenge for a company. The main question at the practical level has to do with the issue of component reuse. If UI components are reusable in other operations, clear savings in effort and time result. Next, an analysis of the typical practices in UI design related operations is given. The analysis includes a summary of the used tools and recognized problem areas in UI design.

Concept design

Concept design has an increasingly important role in modern companies developing and selling personal electronic products. This is because of the fact that the business focus is moving from simple technology advances to customer satisfaction. Products that are easy to use, good looking and offer real value for customers are becoming more important than products offering single technology findings. The focus is increasingly on the right product strategy, segmentation and variation. Concepting is not usually a part of the actual product development, but is organized as a separate process strongly integrating into the company's strategy. It can be a continuous process or an ad-hoc activity managed and controlled usually by general management or by the sales and marketing departments.

The goal of the concept design is to produce product concepts that are further utilized in making the decision for a new product and its development. Additionally, selected concepts work as the initial material for product specification and subsequent development phases. Typically, concept design has three main phases: user study, concept creation and concept evaluation (Ulrich and Eppinger 2000). In a user study, separate user studies can be carried out, for example, with the use of case presentations, product simulations and prototypes. Significant material also comes from product marketing and customer support, which are in constant contact with existing customers. In the concept creation phase, usually a selection of product concepts is carried out. The more concrete the concept presentation is, the easier it is to have valid information from it in the concept study phase. Selected concepts can be further tested with test users, and the concepting process can be iterated several times. After the necessary information has been received, the product decision is made usually by the company management. In this phase, concept or concepts are selected for further development, they are rejected, or they are put on hold.

In general, concepting is a very important phase for visualizations, simulations and prototypes. These are needed in all of the main phases of the concepting process. Based on the experiences of several companies, various means from drawings to slide shows and web animations are used in concepting. The typical tools include Microsoft Office tools with Word, PowerPoint and Visio, Macromedia Flash and Director from Macromedia (http://www.macromedia.com), various animation and drawing programs, and simulation and prototyping tools. In-house tools are also used for concepting quite a lot.

The companies are aiming at an increasing use of functional demonstrations in concepting. Functional and realistic concept presentations can be built even with the previously listed commercial tools, especially if they are extended with in-house and product specific extensions. However, the main question in this approach is whether or not it will be cost-effective for a company to make accurate concept demonstrations created on a case-by-case basis, and how much resources does their developments require? Another question to be addressed is if further benefit can be derived from these demonstrations in other development phases? Generally, building specific demonstrations with various tools does not support component-based development. The tasks are likely to remain isolated, and it is normally not possible to reuse implemented parts in subsequent development phases or even in coming concept designs.

When using the tools listed above a number of further questions also arise. Do the tools support the building of heterogeneous concepts? How are software functionalities presented in the concept? Furthermore, as many of the features in new concepts are based on existing phone features, would it be possible for concept demonstrations to benefit from the existing software components?

Industrial design

Industrial design has a strong link with the desirability and usability of the product. The design strategy is essential when building a brand for a new product. Industrial design co-operates daily with mechanical design, actively with product marketing and increasingly with software development. Product marketing, again, should provide customer feedback for designers. On the other hand, design teams should produce usable product designs for marketing. Often the issue is to maintain the quality and version control of transferred models. The need for active co-operation between industrial design and software development, especially UI design, arises from the fact that both designers and software engineers should be aware of each other's input to a common design target, i.e. the product version/model under development. Designers should be familiar with the software features that might have an impact on specific design decisions, and software engineers should be aware of the mechanical solutions that are available for the product. The situation becomes even more interesting when industrial designers wish to contribute to or guide the UI design, i.e. the visual appearance implemented by the software, such as screen layouts, menus, icons and animations.

Industrial design typically includes two main phases: sketching and detailed design. Sketching visualizes and innovates new designs, forms and product features. Sketching is usually performed parallel with concept design. In detailed design, the design is selected and finalized to fit the specifications coming from other processes. Industrial design utilizes various tools, including 2D/3D CAID and CAD tools, mechanics simulations programmes, drawings and illustration tools for graphics, and rapid prototyping tools. The design methodologies include mechanical design practices in 2D and 3D, test part and mock-up manufacturing, and rapid prototyping.

In general, the communication and co-operation between industrial design and mechanical engineering is fluent. Both of the design areas utilize the basic skills of 2D/3D drawing and visualization, and so they have a long history of active co-operation. Because of this common background, these areas have been improving their co-operation together for several years. As a result, a number of practices for work processes and design tool utilization have been established.

The general challenges met in co-operation between industrial design and software design have to do with the question how to integrate these areas to a common understandable presentation, and thus to enhance the communication between these areas. When industrial designers wish to experience their designs with UI software components, they need, at the minimum level, selected screen layouts to be integrated into their design. However, it has been proven that using only a static layout cannot deliver any detailed information of UI software behaviour to industrial designers. Integrated functional presentation in the form of simulation or prototype is needed, but the question remains how to make these presentations cost-effectively? And further, is there a technology available for creating such heterogeneous presentations? If yes, who should finally prepare the presentations and with which tools?

Product and UI specification

The goal of product specification is to specify the outcome of the development project. As a starting point, the specification uses the information received from product concepting. However, it is not the concepting alone that sets the goals for specification, but the general product requirements are also taken into account. While the requirements can be defined in a separate requirements analysis phase, usually the general requirements are already set by the company's product strategy and product plan.

Specifications are created for the entire product and for its UI software. The main specification defines the overall product outcome regarding, for example, the hardware and software platforms, supported networks (GSM 900/1800/1900, et cetera), other hardware specific features, and a possible product category with style and target customer segments. The main specifications alone set high-level requirements or boundaries for UI specification. The UI must be aware of the available lower-level services and features offered by the hardware and software platform. For example, the screen size, the number of supported colours, the hard keys and buttons available, and the availability of a camera all have a direct effect on UI features. Basically, the overall product requirements will usually serve for fitting the UI requirements to the main product requirements. In other words, when starting with the UI specification, it is already known that certain UI style and applications can be implemented with certain phone specifications.

The effort of UI specification varies a lot from case to case. If a new phone is using an existing UI style, UI platform, and applications, the work effort can be minimized. Another option is to extend an existing phone with new applications or features that require new UI extensions. A third way is to build the entire UI again, which is the most demanding and laborious task. In many companies, new UI styles and platforms are designed as separate projects outside the actual product development, and these styles and platforms are then used over several product models or families. There are, however, also companies that design the UI separately for nearly every new phone model on a case-by-case basis, every time starting from the beginning.

Depending on the extent of the specification work, the tasks in UI specification can include the modelling of interaction flows between the user and phone, defining the new UI and interaction style, defining the behaviour of the selected application or more limited menu feature. Usually, the error and other exceptional situations are also defined. The work procedures and tasks can vary a lot between companies. In some companies the specifications can go to a very detailed level, defining even the exact UI behaviour and appearance. The work can be highly interactive and iterative using different visualization, simulation and prototyping technologies and tools similarly as in the concept design phase. The outcome of the phase typically results in specification documents including textual descriptions, interaction flow diagrams, screen layouts and possible additional material such as simulations and visualizations.

The main challenges in specification concern the communication of specifications both within and outside the specification phase. Different presentations are performed during the specification to demonstrate the ideas inside the team. For example, Microsoft PowerPoint, MacroMedia Flash, and Microsoft Visio presentations are used, and very often from scratch without utilizing any existing material. This leads to wasted effort and time. An even more critical challenge is how to make specifications understandable for subsequent development phases. Even though a lot of effort is put into defining highly detailed specifications with a lot of text, images and interaction flows, these may still remain rather unclear for the UI design and implementation phases. This means that, for example, programmers must work hard to transfer the specification to software specifications to continue with. The question arises if the designing of general product and UI specifications could be done so that the specifications would be easier to understand in general, and so that their structure and level of detail would directly support software design and implementation. It would be worth considering if the specifications could be delivered, for example, as software simulations, which would greatly facilitate these two development tasks.

Usability engineering

As stated already several times in this thesis, good usability has an increasing role in product success. It is the basic feature affecting the user experience that is offered to the user, and as such it strongly influences the purchase decision of the customer. Based on the principles of UCD the main effort in usability engineering is focused on the early development phases including concepting, requirements analysis and specification. Additionally, later usability tests are organized to validate the implemented product decisions.

The valuation of the role of usability engineering varies a lot in different companies. There are some companies that are highly advanced in applying UCD and usability engineering, whereas some companies do not recognize it at all in their strategies or development practices. These differences relate strongly to the size of the company and to the extent of its business on a global scale. The usability process often functions as a guiding process behind the product development, and it is not considered a critical function for the timing or delivery of the product. This means that good usability may be sacrificed in order launch products onto the market in time.

Companies are mainly using in-house tools and practices for usability engineering. The main task is to perform usability tests, and the used tools vary from low-fidelity prototypes, such as paper prototypes, to high-fidelity prototypes and prototype series of the developed product. The degree of realism in tests varies according to need. A test may focus on some specific product parts or features, such as the UI logic, functionality of mechanical controls, or on the overall product appearance and functionality. Usability test are conducted internally and with external test groups. Successful usability engineering requires close and iterative communication with other product development teams and also with the marketing teams. Appropriate means of communication and suitable work processes are significant in implementing usability engineering.

Putting UCD and usability engineering into practice in a company is a challenging task. Usability is a new engineering area and there are hardly any commercial tools available supporting it, especially as regards personal electronic products such as mobile phones. This means that companies must spend a lot of effort building and maintaining their own tools. Very simple, but still efficient methods, such as paper prototyping are sill used widely. However, when products become more complex in their functions, paper prototyping becomes tedious to use, as controlling and building complex “yellow stick” paper simulations is slow and laborious. There is also the challenge of how to test a user experience that is a summary of different contributors such as network operator, service provider and phone manufacturer. The challenges presented by the testing of a multi-contributor experience chain have been addressed by Kerttula and Tokkonen (2002).

Another great challenge can be found in the creation of usability test material. Some companies even wish to benefit from highly realistic prototypes in early user tests, for example, in integrating a new application to an existing phone. Here, low-fidelity prototypes do not work satisfactorily, which leads to the question of how to efficiently build high-fidelity prototypes for this purpose. Building of advanced prototypes for tests can, however, be regarded as laborious and extra work. Thus, to minimize the effort needed, existing material and components should be utilized to as great an example as possible.

Localization

Product localization takes care of producing different language versions of the product, documentation and other supporting material, such as web presentations and marketing material. In its wider extent it also includes the customization of product features for local preferences. The recognition and fulfilling of local preferences has become an important competitive factor on the global markets, especially for mobile phones. The localization of the phone can vary, from the selecting of the right words and icons for the correct cultures and districts, to the development of fully customized applications and other product features.

When analyzing different case companies, the understanding of the role of localization also varies a lot within them. Some companies do not see it as important at all for their business[30]. Many companies have grown to become an international business gradually, which means that the localization has evolved gradually into their practices. This on the other hand, has often led to maintaining old systems that do not serve optimally any more for the latest operational and business needs. For example, a centralized localization material repository including all of the needed and accepted texts, their translations and other resources such as icons and fonts, can be missing. The

[30] The companies in which localization was not recognized as separate product development task were usually strong players on local markets. It was quite clear that even if they had strong plans for international markets, they still had a long way to go in their operations to support that goal.

last category of companies includes those that have understood the role of localization and have planned their practices in line with this. They also often have integrated systems that support the reuse of localized material over different tasks, projects and products.

Product localization is a demanding task, especially regarding the number of supported languages. Localization work is usually performed in a distributed environment, where language specialists are located around the world. Fluent and iterative communication is needed between the UI developers, technical writers, marketing department and customer support persons, and localization specialists in the process. The work is performed with various tools. There are some commercial tools supporting, especially, the translation of web material and PC software applications[31], but for mobile phone UI translation there are hardly any commercial tools available. Many in-house tools and systems have been developed to support the work and the process. Some basic tools making use of spreadsheets, company-wide text and resource repositories, drawings and simulation are also used.

With mobile phones, the UI has a central role in the localization process. The UI texts need to be translated into different languages, and this can be very difficult without being able to view and understand the behaviour of the actual UI feature. It will be beneficial for the translator to see the context of the text as it is in a real product. In practice, another large problem for localization is the order of the tasks in the development project. Often localization can start only when the first UI version is out of the UI design. This version is implemented with a default language, such as English. During the translation it can happen that the text place holders on screens are too small for the text in another language. This results in additional iterations between the UI designer and translator.

---

[31] Examples of commercial localization tools are Alchemy CATALYST from Alchemy Software Development Ltd. (http://www.alchemysoftware.ie), Multilizer from Multilizer Inc. (http://www.multilizer.com) and Visual Localize from AIT – Applied Information Technologies AG (http://www.visloc.com/).

The question arise to what extent and how UI prototypes can be used to facilitate meeting the challenges presented above? Will it be possible for a translator to see and experience the real UI behaviour and the context in his or her work with the help of suitable prototypes? Additionally, would there be a way to make, at least a preliminary language check of the UI design with the needed languages already in the UI design phase, so that possible overflows in placeholders could be recognized earlier.

User documentation

The documentation task produces different product related documents for various purposes. It outputs user guides, and material for customer support, marketing and sales. A lot of documentation material is still produced in printed form, but the role of electronic documentation is becoming increasingly important. For example, a mobile phone package usually includes a CD-ROM with interactive product tutorials and guides.

The documentation makes use of material from different development tasks. It can use the product specifications coming from the specification phase and product visualizations in different image formats coming from industrial design. The UI features and their related resources, such as screen images, icons and fonts all come from UI design. Localization outputs for the different language versions are also used for the documents. The entire process of documentation is usually managed under the marketing and sales organization. Also customer support can specify their specific needs for documents needed in customer support.

The documentation uses various tools, such as word processors, publishing programmes, web authoring tools, and drawing tools, et cetera. In-house built systems are also used for supporting direct transfer of material from other tasks to the documentation, for example. The documentation is usually performed in a distributed environment. Close and iterative communication is needed between technical writers, UI designers, localizations specialist, marketing and customer support persons.

The challenges in this work are similar to those in localization. It can be difficult to understand product specifications and write guiding texts if there is no functional reference product available. If there is a need to make use of user documentation from real UI screen printouts coming from UI design, UI components will generally have to be finalized first. In addition, creating illustrations for documents and possible interactive simulations is laborious already as such, and when taking into account the different languages, the task becomes even more complex. If accurate interactive simulation is needed, then nearly the entire logic and appearance of UI must be rebuilt with a selected tool, such as MacroMedia Flash.

Product marketing and sales

Product marketing often has a controlling role over the entire product development project. It is, after all, responsible for the general success of the project and product. It includes operations that are not restricted only to the actual project period, but are performed as preparation and follow-up operations. Operations such as market and user studies, concepting, and requirements analysis are often performed directly under the marketing department. As a part of the UCD activities, user and market data is continuously collected from external sources and from direct customer feedback. User tests are used for collecting user preferences and for defining new products. During the product development, marketing takes care of the production of the marketing materials, preparations for the product launch, release and promotion activities, and sales channels.

Marketing is often operating in a beforehand-mode. Products are launched in advance, even a half-year prior to their delivery. This puts pressure on the marketing material. How to demonstrate and prepare the material for a product that is not ready yet? Demonstrations are needed, for example, at exhibitions and on the Web, in electronic format. The possibilities of the Web in sales and marketing are further discussed, for example, in (Kerttula et al. 1997b). Product projects run a very tight schedule and after the product launch any support material for the phone must also be made available fast, for example, on the Web.

Since the mobile phone is a software-focused product, a major part of the marketing activities, such as product demonstrations, are related to the product’s software features. Presenting the software features with a means other than by software is difficult. If the marketing department does not have a real product or prototype series available, it usually builds product prototypes and simulations with different tools. However, it should be noted that it is a prerequisite for realistic prototype building that the product specifications should already be set at that point, at least regarding the features that are to be demonstrated. Often, however, the specifications are not fixed prior to something being released from the implementation phase.

When the marketing department builds its own specific simulations and prototypes, this is usually done as additional separate work. The logic and appearance of UI must be built again and no material from the actual UI development is utilized. The documentation of the work will become even more complex when different languages must be taken into account. Even as the product is finalized and available on the market, the need for demonstrations and simulations in an electronic form remains. They are needed, for example, in web marketing and sales, in which the customers can view, try and buy the products. The challenge of creating these presentations effectively remains. The question arises if such presentations should be built as separate work or if an existing functional specification or a UI design from R&D could be reused here?

Customer support

Customer support for a specific product model is usually initiated publicly right after the product is released on the market. There remain a lot of preparations to be made prior to this date. The support includes activities such as preparing support material for the web, training distributors for technical support and service, and testing selected 3rd party services with the phone.

In general, the role of the Web is increasing also as a support channel. The Web offers a 24 hour – 7 days a week distribution channel with global access. It is highly cost-effective, and it only needs to be designed and installed once, possibly replicated, and then it is ready for service. Compared, for example, to a call centre support, an Internet help desk brings direct savings in terms of resources and personnel costs.

The questions and challenges in customer support are basically comparable to those in product marketing. Customer support needs to present and demonstrate software-based functionalities that are difficult to clearly demonstrate without software simulations. Simulations are needed, in which the next issue is how to produce them effectively by reusing as much as possible material from other development phases. A further question is how to handle the different language versions? Additionally, for marketing and customer support purposes, the used simulations must be of very good quality, i.e., the appearance must be appropriate, and the functions and interaction highly realistic.

If there is no understanding and planning of UI software development and its interfacing processes, the company may face a significant loss in effort and time-to-market. Based on the experience gained in the field, there are still many companies that are still employing a basic sequential development process. These companies are highly implementation focused and do not usually recognize the principles of UCD. For example, the role of usability engineering is often undefined. However, there are also natural reasons for this operational model. If a phone manufacturer operates, for example, in a strong relationship with a dominant network operator, the product specification can come directly from the operator. This makes the manufacturer focus only on the implementation of the project. It is the responsibility of the operator to ensure the success of the specifications. In this operation mode, the role of the phone manufacturer even with its own brand comes close to OEM and ODM companies. A sequential process is also typical for smaller manufacturers that make almost all of their phone models specifically for different hardware platforms and do not reuse software and other material. For example, in Asia/Pacific and China there are many mobile phone manufacturers operating in this way [32].

---

[32] Gartner reports in "Established Handset Vendors Face Challenge From Asia/Pacific Newcomers, 24 February 2005" that in Asia/Pacific there are over 90 mobile phone manufacturers. The China Academy of Telecommunications under the Ministry of Information Industry (MII) reports that the number of mobile phone manufacturers is expected to exceed 60 in China in 2005.

The sequential development process of mobile phone UI is clarified in Figure 21. In this case, we may assume that there is hardly any reuse of UI software or UI simulation components over the sub-process boundaries. The process results estimated a development time of 15 months for a new phone model in an optimal situation. Concept design is performed as a separate pre-process. The unclear role of usability engineering is illustrated with a fuzzy 'usability cloud'. If there are any extra corrective UI design cycles coming after the first frozen UI software, the schedule may be extended significantly as these corrections may result in corrective actions also in all the four post-processes.

7.3.3 Virtual design process for UI development

Virtual design aims for shorter time-to-market with minimized effort. This goal is supported with the parallelization of design tasks and effective component reuse. When modifying the previous UI development process presented in Figure 21 to conform with the virtual design approach, the work is started with the following basic settings:

- The focus of the entire process is shifted to the early phases, i.e. concept phase and specification including industrial design aspects.
- Functional simulations coming from early phases are reused as specifications for all the other processes.
- Existing simulations and UI software components are reused in all the phases, not only in specific sub-processes but also between them.

With these assumptions the sequential design process introduced in Figure 21 can be transformed to a virtual design process model illustrated in Figure 22. The presented process model shows an implementation of the basic virtual design process model introduced in Chapter 7.3 in the dimensions of product maturity and elapsed development time.

The transformation of a basic sequential process to a virtual design process offers major advantages. The tasks that earlier had to wait for results from UI software development can now start immediately after the functional specifications are completed. In practice, this means a strong parallelization of these tasks, which again results in significant timesaving in total product development time. Usually, additional savings are reached as the process emphasizing the role of the early phases can recognize possible errors and problems earlier and thus avoid costly correction cycles in the later phases. In our example case, the first customer deliveries can be made 3 months earlier than with the sequential process. And, it is noteworthy that marketing and customer support materials including functional product simulations, localized versions, and product documentation are available even earlier. Next, the process model is described by presenting each sub-process and its implementation with the virtual design approach. The different sub-processes are also demonstrated using the results gained in real customer cases.

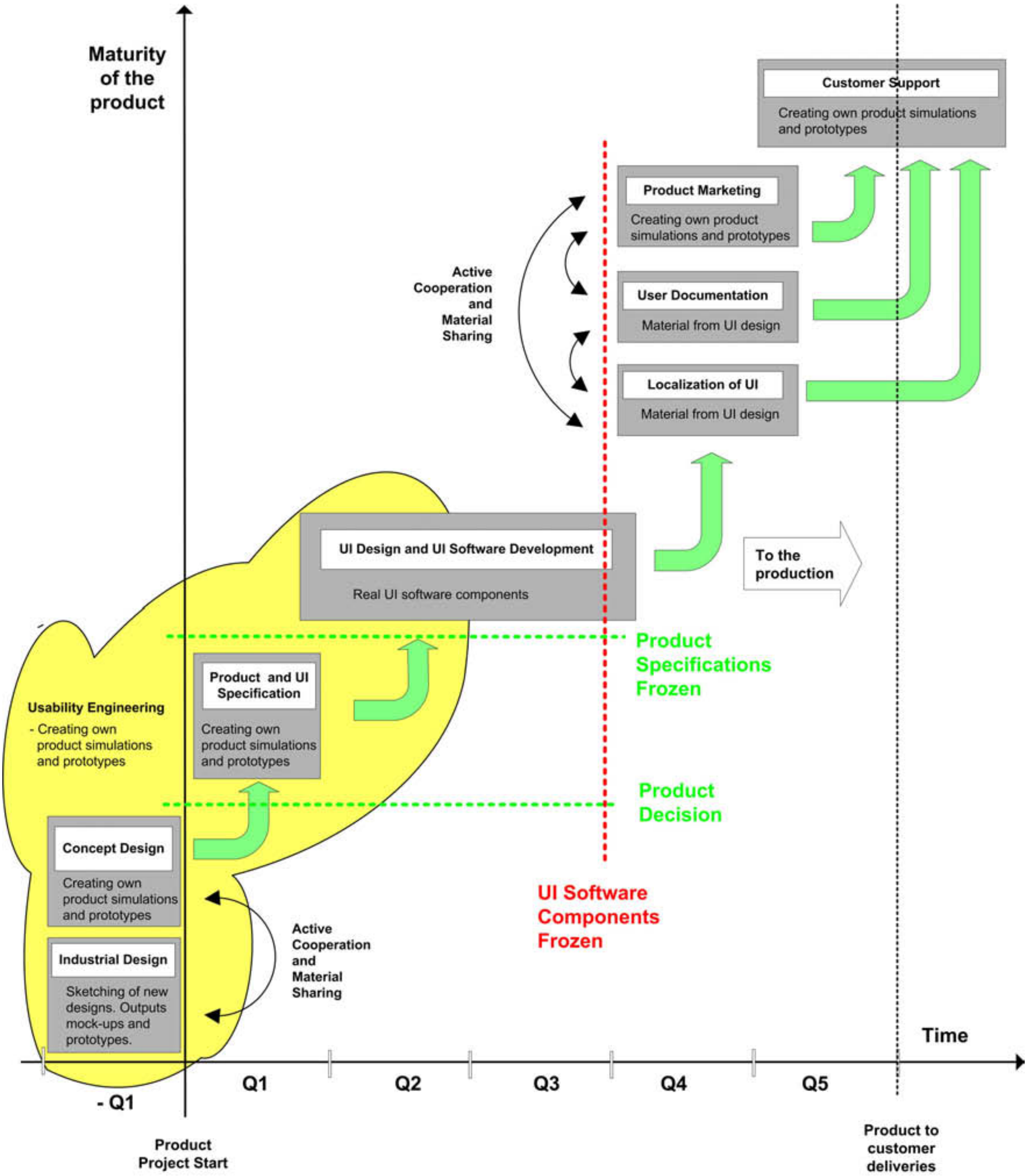


Figure 21. An example of sequential UI software development process without virtual design.

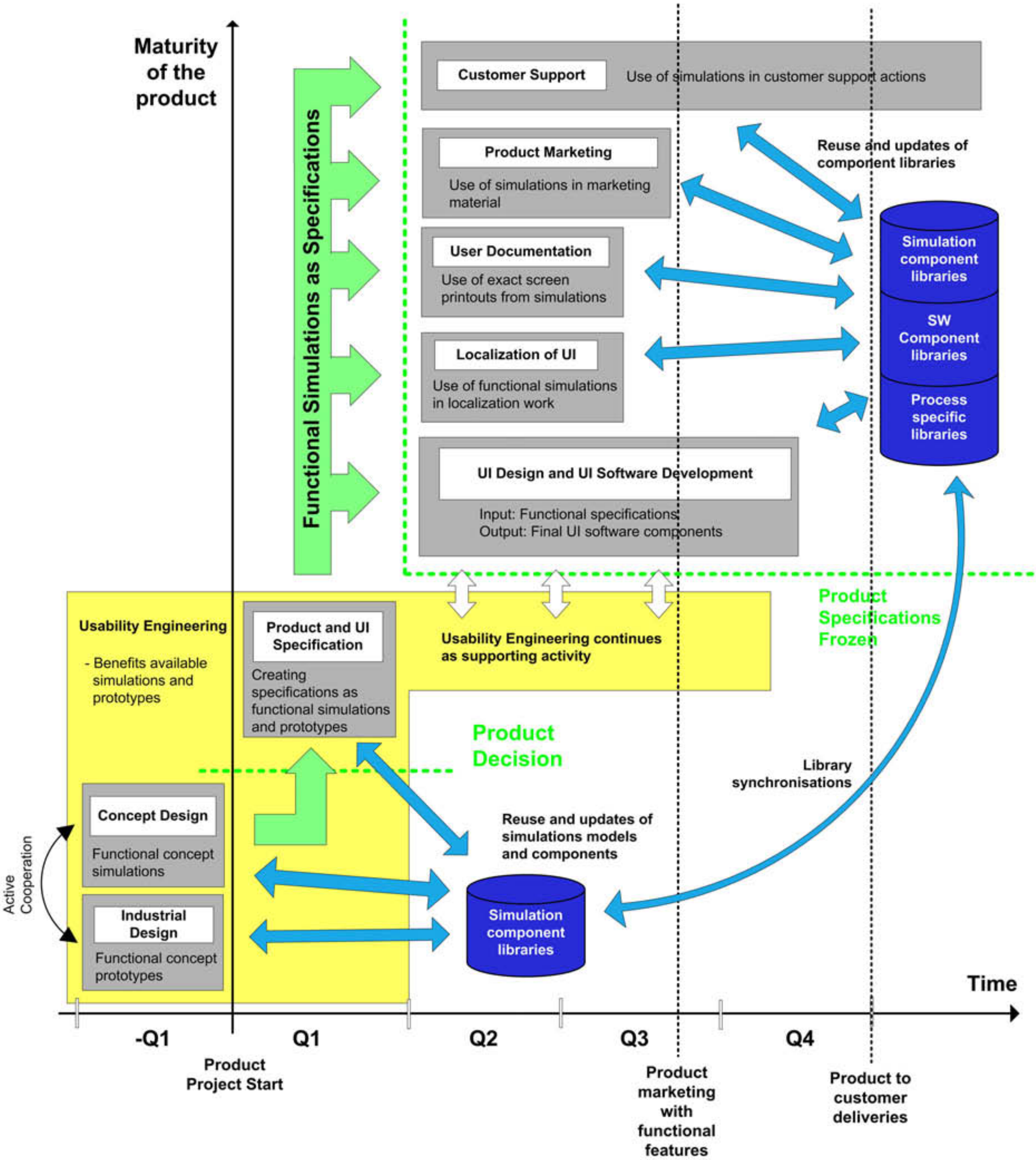


Figure 22. UI software development process for mobile phones based on virtual design.

Concept design

In virtual design, product concepts are implemented as virtual prototypes that can provide a realistic presentation of the target product already in the early design phases. The idea is to use heterogeneous prototypes for integrating real product components and simulated features. Existing simulations and simulation components are reused to enable faster concept creation. If new components are built during the concepting they can be used further in subsequent development phases and in new concept projects.

The product concepts have been implemented with Cybelius Maestro in several customer projects. Either more focused concepts, such as rapid modelling of the UI main structure without programming, or more extensive concept simulations integrating real product components from earlier products have been implemented. One example is the Nokia RFID (Radio Frequency Identification) tag concept, which was created to demonstrate the possibilities of near field communication[33] (NFC) with mobile phones (Repo et al. 2005). Later the concept was also used to find out the right interaction sequences used with NFC technology. The first ideas regarding the Nokia RFID device were developed around a separate device for reading data from and writing data to the RFID tags (Figure 23). The concept was then further developed so that the reader would be integrated into the mobile phone.

One important motivation for Nokia to use Maestro in the RFID concept modelling was the availability of Nokia Series 60 UI simulation components. These were provided for Nokia as a part of Maestro Nokia Series 60 EP. In addition, the Maestro tool was selected, because Nokia wanted to include real GSM phone functionalities in the prototype. This was implemented by connecting a Maestro simulation to a GSM card phone attached to a PC. The RFID prototype was able to send and receive real SMS messages, such as purchasing a ticket for the Helsinki metro.

---

[33] More information about NFC and its applications can be found at http://www.nfc-forum.org.

The created Nokia RFID prototypes had an important role in demonstrating the idea and capabilities of NFC. Maestro and the prototypes created with it were used in two major concepting phases, which led to the proposal of the RFID tag reader as an embedded feature of a mobile phone. Nokia launched the first product as a NFC enabled shell to the Nokia 3220[34] (http://www.nokia.com) phone, which is one of the first ubiquitous computing products already introduced to the consumer markets.

Industrial design

Virtual prototypes help industrial designers to integrate their work with concept design. Product designs, such as covers and accessories, are shown as a part of functional and interactive product presentations, which provides a more realistic view of a new product. Virtual prototypes can be viewed on a PC screen or integrated into a separate hardware mock-up providing an even more realistic feeling of the industrial design in the form of a mock-up design.

In the Maestro customer cases, the integration of industrial design to Maestro prototypes is an everyday operation. Different company specific product designs, such as all the PUI models of a company's phone models, can be integrated into the company specific project templates used in Maestro. New designs can be imported to Maestro projects simply by importing design images from industrial design tools to Maestro Builder's resource model.

[34] See press release "Nokia Unveils the world's first NFC product - Nokia NFC shell for Nokia 3220 phone" launched November 2, 2004 from www.nokia.com.

Figure 23. Nokia RFID concept model prototyped with Cybelius Maestro (image courtesy of Nokia).

Product and UI specification

After a product decision has been made, a more detailed specification cycle is performed and the virtual architecture of the product is modified to conform to the real implementation architecture. The GUI components of the product are selected and specified. For example, a part of the GUI components is taken from earlier product versions, and the rest are specified as new components. The selected industrial design is attached to the specifications to define the screen size, keys and other possible mechanical controls. The behaviour of the main UI and different applications are defined by using different techniques, such as interaction flows. After this, the UI logic is modelled and turned into functional prototypes. The accuracy of virtual prototypes is kept high enough for them to be able to exactly define the appearance and behaviour of the UI and UI controlled functions of the product. With this approach, the frozen specifications presented as virtual prototypes can be reused directly in subsequent process phases.

Many companies (ME, MJ, MU, HJ and PE in Table 1) have utilized Cybelius Maestro in product specification. The typical way for its use is to build product simulations to support other specification material. It has been noticed that functional and interactive prototypes can be used to demonstrate concretely and clearly a number of software-based features that would be rather laborious to define and present with other means. There were also some time-critical projects, in which Maestro simulations were used as such as a specification for the implementation performed by a subcontractor.

As described in Chapter 7.3.2, it is a great challenge for specification work how to interpret the specifications and continue implementation from them. Even though there is a massive amount of specification material available, including, e.g., text descriptions and screen visualizations, it is difficult for implementation to understand and reuse them. This problem has also been recognized by some Maestro customers that are in a process of making Maestro their main specification tool. The use of Maestro may set challenges for traditional specifiers, who are not necessarily familiar with specifying product functions as exhaustively as needed for programming. However, when more detailed specifications, including state machine diagrams of behaviour, have been prepared in the specification phase, they can be directly utilized in implementation. Additionally, if a mobile phone platform specific architecture is brought to a Maestro specification, project specifiers can be guided to even more implementation-friendly work producing specifications conforming to the target platform architecture and, for example, including the API calls offered by it.

Usability engineering

Usability tests are done already in the concept and specification phases. They can directly employ the concept designs implemented as simulations and prototypes. The goal is to have the main product ideas and features tested prior to implementation. With this approach, virtual design supports a forward-looking role of usability engineering. As the specifications are tested before starting the implementation, there is a chance to filter out the first usability related problems in specifications and thus also to avoid the first redesign or update cycles. After being used for supporting the concepting and specification tasks, usability activities are carried on in guiding and controlling UI software design and development. In practice, usability tests are performed with virtual prototypes, which can provide the user with experience comparable to real products. The focus of the tests may vary according to their objectives. Either extensive test arrangements including the use of hardware mock-ups can be used, or only 2D simulations of UI screen graphics.

Cybelius Maestro simulations support usability testing in several ways. Some companies are using basic Maestro simulations running on a PC screen in user tests. The simulation can be an advanced product simulation presenting almost the final appearance of the product. In some cases, for example, when UI logic is tested, the detailed design of the product has been hidden. Maestro simulations have been also used as a substitute for paper prototypes when complex product features have been tested. The experience from these test sessions has been encouraging. When an experienced Maestro user has been participating in a test session, it has been possible to change the behaviour and appearance of the simulation even quickly during these tests.

More advanced usability tests have been conducted with hardware mock-ups and real mobile phones. For example, the idea of Cybeo ONE mock-up arrangement has been transferred to a Bluetooth-enabled Symbian phone. The client software of the phone connects to Maestro simulation running on a PC via Bluetooth and transfers the simulation screens to the phone and, vice versa, the key-events back to the PC. With these arrangements, it has been possible to test new mobile applications highly realistically.

Localization

Localization of UI resources can start immediately when functional specifications are available. The localization specialists can see and understand the context of different terms better when simulations are available. Existing material is used as much as possible from component databases, and only the localization of new UI functions and applications is needed. Usually, localization can utilise a specific localization database containing the accepted material for all languages and UI resources.

Cybelius Maestro includes a default support for localization. Each simulation project can use the localized UI resources defined in the Resource Model. This means that the translation of UI resources can be performed during the UI design in Maestro. Additional components used for connecting to a company specific localization database have also been implemented. With this arrangement, it is possible for a Maestro user to fetch the official translated resources from a company database. The feature that checks automatically the fitting of translated text resources to their placeholders on screen components has also been recognized as an important feature by Maestro customers.

User documentation

Technical writers and other documentation experts can start their work based on functional specifications. They do not need to wait for results from the implementation. The frozen specifications including detailed UI functions, screen component and layouts can be used as such in the user documentation. The amount of duplicated work needed to produce UI screen images with other drawing tools can be reduced. Simulations, on the other hand, can be integrated as such into electronic documents, such as CD-ROMs.

The quality and accuracy of the graphical UI components simulated with Cybelius Maestro usually equal to those of a real device. For example, in Maestro, the GUI components and resources, such as icons and fonts, of the Nokia Series 60 EP are exactly the same as used in the real Nokia Series 60 UI platform and phones based on it. The screen layouts and graphical components can be copied or exported directly from Maestro Builder to documentation use. Furthermore, thanks to Maestro also supporting design-time localization of the resources, it is possible to export screen graphics in different languages to documents. Besides the basic cut-and-paste documentation support, Maestro simulations can be directly utilized in electronic documentation as JAR files, which pack Maestro simulation and required resources, like localized texts and graphics, into a single executable Java application file.

Product marketing and sales

As soon as functional specifications are available, they can be utilized in product marketing. Applying functional and interactive simulations in marketing will provide a more complete product experience for potential customers. Especially software features that are difficult to explain or illustrate can be demonstrated more efficiently. Marketing simulations can focus only on the selected features of mobile phones or on more extensive simulations showing the whole user experience related to the integrative use of the phone and its value-added services. When simulation material based on functional specifications is used, the need for building specific simulations for marketing use is reduced significantly. This results in distinct savings in effort and time.

When Maestro simulations are created, a lightweight Java Applet version can be generated from them for web use. More complex Maestro simulations including real hardware can also be utilized, for example, in exhibitions and on special demonstration premises. One and the same product simulation can be used for different market areas such as UI localization features in Maestro, enabling an easy selection of the UI languages for simulation execution. When creating web pages for different countries, only the country specific web material around the Maestro Applet simulations needs to be designed.

An example of using a Maestro simulation in product marketing and sales is given by Polar Electro (http://www.polar.fi). The detailed modelling of their new AXN700 heart-rate monitor (Figure 24) was performed with Maestro, and the simulation has also been used in the marketing and in the training of their global resellers. The simulation was also localized for two languages.

Figure 24. Cybelius Maestro simulation of Polar AXN700 (image courtesy of Polar Electro, Finland).

Customer support

Similarly to product marketing, customer support can utilize the functional and interactive simulations based on functional specifications. The effort needed to modify the specifications for customer support depends on the used base technology and available tools. If specifications are done so that they can also be executed on the web, or if there are tools available for converting them to web-based simulations, they can also be utilized in web-based customer support. The so-called e-learning web pages can be opened for different phone models. Considering that localization support for simulations will be finalized parallel to customer support material, e-learning pages for different countries can be launched simultaneously.

The Java Applet simulations of a mobile phone provided by Cybelius Maestro offer a practical way of implementing e-learning support on the Web. The need for additional work is minimized as the simulations from the specification can be used directly. Like in product marketing, e-learning simulations can be immediately produced for several languages. One Maestro customer has produced e-learning simulations with Maestro for their global web pages and used some simulations with over 10 different languages.

UI design and UI software development

Even though the dominance of the more traditional UI design and UI software development phase is reduced in the context of the new virtual design based process (Figure 22), it still remains a very important and critical phase for the total performance of the development process. Especially when a lot of new features are included in the mobile phone model under development, the effort and time needed for UI software development becomes significant. The general advantage of the virtual design approach for this phase comes from component reuse. Functional specifications, especially when prepared well, already specify the places in the design where existing components are usable and what possible modifications are needed for them. Appropriate specification also offers real components for the development phase. For example, graphical UI elements can come directly from specifications, which means that this part of the work is, in practice, moved from traditional UI software development to the specification phase. If the specifications already recognize and follow the planned product and UI architecture, the implementation work in the UI development phase is in general much more straightforward, which means savings in time and effort.

New software components and new versions of existing components are developed with suitable software engineering practices and tools. If available, such tools should be used that generate real target code or even compile executable binary or byte code for the target system directly from the simulation models. This possibility emphasizes even more the role of simulation components and models in the UI software development phase. Finally, when new components, either as simulation components or real target code components, are added to component databases, they are again reusable in the next development projects.

During the last two years, Cybelius Maestro has been extended also to provide support for the implementation phase. The idea of producing real target code from visual design models supports the general approach of RAD (Rapid Application Development), which aims at faster program development with more easy-to-use tools. The Maestro code generation has been demonstrated for several platforms. Already in the 3GSM World Exhibition 2004, Cybelius demonstrated Nokia Series 60 resource and skeleton code generation for Symbian OS directly from Maestro Builder projects. After that code generation has been extended with other C/C++ target platforms, especially in Japan (Cybelius Software 2005b). For example, a Linux code generation for GTK+ UI toolkit has been demonstrated. The latest code generation EPs have been demonstrated for the Java Micro Edition (JavaME) platform including Java CDC and Java CLDC application generation.

7.3.4 Analysis of the case

This summative case study of mobile phone UI development provides important information on the possibilities of virtual design. It shows that when applied correctly virtual design can shorten the development times by parallelizing the design tasks and it can reduce development costs by reusing components within different tasks and product projects. Even though the case presents a generalized study and not a complete implementation of specific product projects with virtual design, it demonstrates various specific sub-processes and a co-operation of certain sub-processes in which virtual design can be of benefit. Based on the experience from these separate design areas, it can be concluded that virtual design can be successfully used for supporting a complete product design process in the future. The prerequisites for this are that the companies understand the potential of the approach and that there are suitable tools available for putting the approach into practice.

The experience from Cybelius customer cases shows that the major obstacles for a more wide-spread adoption of virtual design are mainly in the organizational issues. The shift to virtual design may be too difficult for a company to arrange, especially if the existing processes are running satisfactorily and the business pace does not allow any time for introducing major changes to existing practices.

Another organizational challenge comes from the fact that departments, design groups and sub-processes tend to be selfish. They often look at changes only from their own point of view, and cannot see the overall improvements in company performance. Putting virtual design into practice is bound to lead to changes in existing working methods and learning new practices. For some design groups, such as specifiers, it can even mean more work or more demanding tasks. To overcome these organizational challenges, a company must have strong management to see the need for and the benefits of virtual design for the company in the longer run.

In planning the implementation of virtual design, it is important to understand the role of company specific components and templates. The availability of good basic components is essential for a successful rollout of virtual design. If there are no ready-made components or templates for the products that the company is working with, it can be too laborious for different groups to start developing and using virtual prototypes. For example, if usability engineers or localization specialist cannot have prototypes or their components from a common component repository or from other design phases, they usually do not have the time or expertise to develop their own prototypes from scratch. To avoid these problems, it is important that the management organize the development of basic company specific components and templates, before virtual design is introduced for wider use. This is important especially if the tools, such as Maestro, are started to be utilized by different groups with different timing and without waiting for the basic material to come from the concepting and specification phases. In many companies, it is specially concepting and specification that take care of the development of the first prototype components for a common repository. But even then it must be noticed that if there are no components available, their development in the first round takes extra effort and time, regardless of who is taking care of that.

Before leaving this summative case study, it is worth noting that the presented UI development process (Figures 21 and 22) does not present any complete mobile phone design process or project. The design areas connected to the UI development are not described completely here, but they are examined especially from the point of view of UI development. Usually, a real development project of a mobile phone is a summary of several sub-processes, such as hardware, mechanical, and software design, progressing in parallel towards the implementation of a final product. Depending on the company and its practices, the UI development can be recognized as an important, separate development area, or implemented as an embedded part of other software development processes without any special attention paid to it. One view of real mobile phone development is given by Ketola (2002), who describes the processes and practices used at Nokia. Ketola shows that Nokia uses a sequential milestone-based control approach for development areas progressing in parallel. UI design is one of these separate development areas, in addition to concept design, industrial design, mechanical design, software design, hardware design, localization and user documentation.

8. Analysis and discussion

The results of this thesis are summarized and analyzed in this chapter. First, the overall evaluation of the research and the selected research method is done. Then the results related to the main research phases are examined, and the answers to the research questions are given. After this the contribution of the thesis both for academic and industrial communities is discussed. Next, some important considerations on putting virtual design into practice are presented, and finally, the chapter is completed with a short discussion of future research directions.

This thesis introduces virtual design as a new product development and business process framework for companies developing personal electronic products. The goal of this thesis was to study how virtual design and virtual prototypes could be of benefit to the development of personal electronic products. The research represents constructive research in technology and design science. Additionally, it implements applied research with a strong connection to practical industrial cases. The research was conducted in three phases as is typical of constructive research:

1. Analysis phase: The analysis of current practices and challenges in the development of electronic products was done based on the experiences gained from several industrial design cases. The requirements for better product development practices were also defined based on the experiences from the case companies.
2. Construction phase: In this phase the virtual design and virtual prototypes were defined as constructs or innovations of the research. They represent methods as a construction (March and Smith 1995). Furthermore, instantiations of these constructs were also implemented in the presented design cases.
3. Evaluation phase: The proposed solution, a virtual design process model relying on virtual prototypes, was evaluated against the typical evaluation criteria used in constructive research.

The evaluation of the results in constructive research is based on the following basic question: Can we build a certain innovation and is it useful? March and Smith (1995), and Järvinen (1999) further elaborate these questions for different innovation types. For method, the authors set such criteria as operationality, efficiency and ease of use. For the instantiations, the typical criteria are efficiency, impact on the environment and its users, and cost/benefit ratio. Comparing the results of this thesis against the typical evaluation criteria, the following conclusions can be drawn:

- Method innovations of virtual prototype and virtual design were constructed and set operationally.
- Instantiations of the virtual prototypes and virtual design were constructed partially and evaluated in design studies.
- The evaluations of the instantiations showed that these could improve the performance of the companies in certain areas.

Based on the results and the principles of the qualitative research, it can be further deduced that virtual design can offer improved and more effective working practices to the industry also when virtual design is applied in full scale.

The selected research method of constructive research performed well in this work. The focus of the thesis was on applied research contributing strongly to industry. Industry has thus been functioning as the main source of empirical data used in the analysis phase of the work, and providing a test field for the evaluation phase. Taking into account the hectic speed of evolution of the target industry and the innovativeness and extent of the virtual design approach, it would have been impossible to apply any other research method within a reasonable time-scale. For example, it would have been impossible to conduct any detailed and meticulously prepared and planned case studies of the virtual design implementation in several case companies within a 2- or 3-year schedule.

Even though virtual design offered a highly interesting research subject, its novelty and wide scope caused great challenges for the research work. Especially at the beginning of the research, before 2000, it was difficult to convince companies of the idea of virtual design. In practice, the most important customer cases were started only after 2001, and the prerequisites for the success of this work were the active research and promotion period between 1998 and 2001 at VTT and the subsequent international sales activities at Cybelius Software. Soon after the first real design cases were started, it was noticed that testing the full scope of the virtual design in companies would take several years. The companies had different starting points, and some companies wanted to apply virtual design principles in early design phases whereas other companies were more interested in implementation support or after-sales activities. It ultimately turned out that gaining experience on full-scale enactment of virtual design was impossible within this research work. Similarly, it was only a dream to get statistical figures on the performance of virtual design in terms of required effort, development time and quality when compared to earlier product development projects using more traditional development processes. If available, this kind of performance measurements would, of course, have further increased the value of the thesis.

8.1 Summary of the results

The results of this thesis can be summarized in eight subcategories:

1. *The results of analyzing the business area of personal electronics.*

Personal electronic products have a relatively short history. The role of user-centred and software-focused personal devices has increased in the industry of electronics and telecommunications since the beginning of the 1990s. The business area of personal electronics and its challenges can be described with 4 important characteristics. First, in personal electronic products the role of software is very important. Even though they are still categorized as electronic products, the major part of the effort and costs in their development is caused by software related work. This means that many manufacturers of electronic devices have, in fact, been shifting towards a software business.

The second characteristic of the personal electronics industry is the role of the end-user and customer satisfaction. With personal electronics the customer is the king. Personal electronics get closer to the customer than many other electronic products. Often the users carry their devices with them all day, and personalize them according to their private tasks and preferences. When there are many products competing for user’s attention, it is clear that good usability and an appealing design are key competitive factors in the markets.

The third characteristic that describes the demanding nature of the personal electronic business is product development practices. Personal electronic products are multi-disciplinary products requiring expertise from many areas. It is not only hardware, software and mechanical engineering that work together, but also new expertise areas, such as industrial design, usability engineering, and product localization are needed for the business. The increasing number of participating expert groups makes the development of personal electronic products a complex task in the sense of effective communication and co-operation.

The last characteristic describing the personal electronics business is internationality, which applies both to its markets and development organizations. Personal electronic products are usually targeted for global consumer markets, which sets interesting demands on their development. Unlike simple mass products, personal devices require localization to take into account the preferences of different cultures and user groups. For product development, international operations suggest that the development work is usually done in distributed organizations, which set new demands for work processes and communication.

2. *The results of analyzing the typical product development practices and tools used in the electronics industry.*

When analysing the typical product development practices used in companies that are developing electronic and telecommunication products, it is important to see that many of the them still apply the process models coming from either the software engineering or systems engineering area. This is likely to lead to problems, as both of the approaches have limitations in the context of the development of personal electronic products. The challenge in applying software engineering practices to personal electronics lies mostly in the early integration of software engineering with other engineering areas and business operations. Recognizing the importance of the user interface design and good usability, software engineering practices should emphasize the role of user preferences and usability within their process models. Systems engineering practices, again, usually focus on large systems in relatively stable business areas, where the work is typically guided by several standards and other national or international regulations. As such, it is difficult to implement them in areas of rapidly changing technology and market dynamics that are typical for the personal electronics business. Furthermore, systems engineering has no special focus on software development. It regards software as only one component or part of a system. Instead, it should recognize the dominant role of software in the final features of the system, as is the case with personal electronic products. Other, more recent product development models have also been introduced, especially, under the UCD methodology. In these models, the problem is usually that on a practical level they do not provide any guidance on how to integrate existing process models with them, or on how they could be used to cover other design areas that are not defined or recognized in them.

In addition to development process analysis, current development tools, and simulation and prototyping techniques are also studied. The tools are categories in actual design tools, design analysis tools, design support tools and other supporting tools. Different classifications of prototypes have been described and different prototyping approaches have been introduced. In general, the boundaries and definitions for prototyping approaches are fuzzy. This is why the main approaches are summarized in a separate table (Table 2). Virtual prototyping is introduced as an approach that is applied in different design areas with different means and purposes. Related to the study of simulations and prototypes in companies, their role in UCD is also clarified.

3. *The results of analysing the problems in the development of electronic products from the perspective of personal electronic products.*

The problems that are faced when typical development practices used in electronics industry are applied in the development of personal electronics are explained in Chapter 4. The analysis is performed based on the general deficiencies found in the typical development process models used in the industry, and on the results attained in the case studies between 1996 and 2001. Similarly, the problems related to development tools and prototyping support are identified. Most of the problem areas are demonstrated with examples from case studies. The problems found are listed in Table 3, in Chapter 4.

4. *The results of defining the requirements for a product development process model, development tools and prototyping tools supporting the business of personal electronic products.*

The requirements for better product development practices in the area of personal electronics development are given in Chapter 4. They have been defined by analysing the problems given in Chapter 4.1. Direct feedback from case studies is also taken into account here. First, the general objectives of the product development are summarized in the context of company operations. It is stated that the main goal of product development is to support the company business. Next, the requirements for an enhanced product development process are given, and the supportive means and measures for attaining each requirement are described. After this, the requirements for a better development tool and prototyping support are listed. All of the given requirements are summarized in Table 4 at the end of Chapter 4.

5. *The results of defining virtual prototyping and virtual prototype in the context of personal electronic design.*

The definitions for virtual prototyping and virtual prototype are provided in Chapter 5. Virtual prototyping is defined as a process framework, and virtual prototype as an implementation technology for the previous. The given definitions modify the earlier definitions of virtual prototyping according to the requirements set by software-focused and user-centred products. The definitions result in a flexible prototyping approach that supports multi-disciplinary and heterogeneous prototyping. The construction model for a virtual prototype is given by presenting the product features that it must be able to prototype. The VRP approach is also defined as a sub-category of virtual prototyping.

6. *The results of defining the virtual design and process model to implement it.*

A definition of virtual design, a product development and business process framework utilizing virtual prototypes in different design tasks, is given in Chapter 5. A process for implementing virtual design is explained step-by-step. The presented process framework functions as a starting point for the company specific customization of the process. The main idea of the virtual design is to utilize the same virtual prototype presentation of the target product from concepting to the final marketing and customer support operations. The prototype incrementally evolves towards a real product through substituting simulated product parts with real ones. This aspect of the evolutionary product life cycle support of virtual design is clarified in Table 5, which shows how the evolutionary presentation of a virtual prototype not only provides support for a company developing a product but also for a customer buying and using the product. In Chapter 5.3 a summary of related research in the area of virtual design is also given.

7. *The results of analysing the tool support needed and available for virtual design implementation.*

The tool support for the implementation of virtual design is analysed in Chapter 6. The architectural approaches to implementing a virtual design process model are studied. Two approaches are suggested. One implements the common virtual design space with a separate tool or system solution that links to other design area specific tools and combines their input into an integrative product presentation. The other approach relies on a strong design area specific tool and tool platform, called a master tool, that covers, at least to some extent, the operation of the virtual design phases. The master tool is given a strong role in the development process and other design areas are integrated into a shared product presentation presented by the master tool. The potential of commercial tools to implement the given architectures is discussed. To summarize, it is difficult to find a single commercial tool or tool platform offering adequately extensive support for virtual design in the area of personal electronics. In practice, to build an extensive tool support for virtual design, several commercial tools must be integrated together, and further complemented with self-made extensions for integration parts and areas that are not supported otherwise. There are, however, some commercial tools that have potential to offer more advanced tool support for virtual design. One of these tools is Cybelius Maestro, which was originally developed to support a virtual design approach. Maestro is introduced in more detail in Chapter 6.4.

8. *The results of applying virtual design in the development of personal electronic products*.

The potential of virtual design in the development of personal electronic products is evaluated by three design studies presented in Chapter 7. Two of these introduce the relatively limited design cases of demonstration concepts called ComPen and Cybeo ONE. However, the cases were not originally meant for the study of virtual design, but their purpose was to study the possibilities of virtual prototyping technologies, and to demonstrate the capabilities of the applied tools, VTT's VRP Environment and Cybelius Maestro. The cases shed light on several important features of virtual prototypes and the tools used, and as such proved their support for virtual design. The results of ComPen and Cybeo ONE cases are reported in Chapters 7.1.2 and 7.2.2.

The third case study introduced the use of a virtual design approach in the development of mobile phone UI software. The case summarized the results from several industrial cases in which Cybelius Software was participating between 2000 and 2005 in Europe, Japan, South Korea and the USA. The case explains the role of UI as an integrative task that has links to many other tasks, including concept design, industrial design, product and UI specification, usability engineering, localization of UI, user documentation, marketing, and customer support. Each of these areas was summarized in terms of typical working practices, tools and challenges. Their co-operation was illustrated with a design flow of a traditional sequential UI design process (Figure 21). This process model is still used in many companies as a foundation of their development work.

Next, the sequential process was reorganized into a concurrent process model implementing virtual design. The reorganized model is presented in Figure 22. This process illustrated a customized version of the basic virtual design process introduced in Figure 10, within Chapter 5.5. Additionally, the customized process model was translated into a dimensional presentation of elapsed time and maturity of the product. The effect of the process change was described for each task linked with the UI design. Finally, the results of using Cybelius Maestro in the different tasks of the customized process model were given. Task executions were illustrated with examples from customer cases. Even though – for confidentiality reasons – no single case applying the presented virtual design process model could be presented in its entirety, several design areas were addressed, and real experience from the industry was presented. It can be assumed based on the individual task experiences that virtual design will also function as a complete process model for products under development.

8.2 Answers to research problems

The research problems and questions presented in Chapter 1.3.1 have been addressed and the following statements can be made regarding them:

> How can virtual design, a product development process framework utilizing virtual prototypes, be used effectively in the development of personal electronic products?

Virtual design can be used as a process framework that is suitable especially for personal electronic products. It recognizes the special requirements that are set by the dynamic and innovative business of personal electronics for effective product development and other business operations linked to it. Virtual design supports the multi-disciplinary co-operation needed in the development of personal electronics. Different design areas can be integrated into a shared product presentation created by virtual prototypes. This shared presentation of the design target supports enhanced product and process visibility. Additionally, the functionality and interactivity of virtual prototypes offer a self-explanatory presentation of the object, along with clarifying the software functionality of the product.

Virtual design emphasizes the importance of effective concepting and concept validation methods. These are needed to ensure the correct product decisions, which are also fundamental as regards customer satisfaction and the desirability of the products. Concepting is supported with functional and interactive virtual prototypes that can create a realistic product presentation and user experience already in the very early design phases. In order to create realistic prototypes fast and cost-effectively, it is important that the prototyping approach supports heterogeneous prototyping and component reuse. Both of these requirements are supported by virtual prototyping.

Virtual design recognizes the central role of software-based product features and software development. Most of the features in personal electronics are software-based. It is usually difficult to demonstrate or illustrate the functions of software before its implementation without the necessary software engineering knowledge. In virtual design, software can be developed using the prototype approach. The functions of the software can be presented and validated before the design and implementation phases as a part of a virtual prototype. The prototyped software parts can be replaced with real implementations later in the process.

Virtual design supports incremental and component-based product development and prototyping. The first product prototypes created already in the concept phase can evolve along the development activities towards a real product implementation. Existing product components are used in the development process as much as possible, i.e., existing real product components are used as a part of a virtual prototype already in the first prototypes (cf. heterogeneous prototypes), and later, according to design decisions, their number can be increased. The component reuse principle is also applied to prototyping components. Existing prototyping components are reused to as great an extent as possible.

Virtual design supports CE with prototype reuse. Companies use prototypes and simulations within many tasks. Usually, these are built with various tools resulting in prototypes and simulations in various form and formats. The results and their components do not usually move from one task to another, or they are not even reused inside the same tasks in subsequent projects concerning the same product. In virtual design, a single prototype and its components can be utilized over different design tasks, and even outside the product development. Of course, prototypes can also be utilized in new product projects and even outside the company by other companies[35]. Prototype reuse enables parallel operations under the term CE, especially if the product specifications can be presented as an accurate virtual prototype after the specification phase. In this case, the virtual prototype can substitute the real product implementation in

[35] This refers, e.g., to a possibility of mobile phone manufacturers to deliver virtual prototypes of their phone models to operators and distributors to be used for marketing and customer support purposes.

various subsequent tasks, which usually have to wait for real implementation results to become available. As other tasks no longer need to wait for the implementation phase to finish, they can be initiated parallel to it. This situation is clarified in Figure 22 in Chapter 7.3.3.

Based on the experience and results gained from several industrial cases (Chapter 7.3), it can be summarized that virtual design can be of benefit in the development of personal electronic products in many ways:

- Virtual design can improve customer satisfaction by enabling more secure product decisions in product concepting by using realistic virtual prototypes.
- Virtual design can shorten the time-to-market by parallelizing the development tasks, marketing and customer support activities.
- Virtual design can cut costs by supporting prototype and prototype component reuse. Additionally, it can help minimize the prototyping tool cost if several tools can be replaced with one that produces virtual prototypes.
- Virtual design can help in achieving better product quality with component-based development and prototyping.

1. What are the limitations of the typical product development process models and tools used in electronics industry, when they are applied within the development of personal electronic products?

The process models and tools typically applied in the development of electronic products are analysed in Chapter 3, and the found limitations are specified in Chapter 4.1. The results of these chapters can be summarized by stating that many process models do not recognize the new requirements specific to personal electronics design. In other words, they do not recognize the central role of software and UI in these products. Neither do they not recognize the importance of effective concepting for business in general. They also fail to recognize the multi-disciplinary nature of the development. Additionally, they do not understand the potential of effective prototyping in process support. In general, many traditional process models are not dynamic enough, i.e. flexible and customizable, for the dynamic business of personal electronics.

Development tools share many of the problems of the process models. Many tools cannot recognize other expertise areas, and they are not capable of co-operating over different design areas. Similarly, the commercial prototyping tools available cannot integrate design information from different design and expertise areas. Additionally, prototyping tools do not support fast prototype creation with a heterogeneous prototyping approach, which would be useful especially in the product concepting of personal electronics.

2. What are the requirements for the product development process model and development tools supporting the development of personal electronic products?

The requirements for better product development practices supporting the development of electronic products are given in Chapter 4. When these are examined particularly from the perspective of personal electronics design, a number of issues can be highlighted. Regarding process model requirements, the important issues are:

- Early validation of product concepts, versions and features. This is supported specially with heterogeneous modelling and functional product models, i.e., by applying virtual prototypes and component reuse.
- Efficient communication. This is supported especially with enhanced product and process visibility and functional product models, i.e., by using virtual prototypes.
- Concurrency of the operations, which is supported with enhanced product and process visibility, heterogeneous modelling, functional product models and component reuse.
- Adaptivity and flexibility of the process model, which are supported with tool and technology independence, iterative and incremental development practices and the process customization capability.

Regarding tool requirements highlighted issues are:

- Tools should have flexible interfaces.

- Tools should be customisable for new tasks and technologies.
- Tools should have a flexible architecture and technology base.

Regarding the requirements set for prototyping support, the important issues are:

- Support for cross-engineering product presentation.
- Support of the incremental design process.
- Support for heterogeneous prototyping.

3. How can the virtual design process model utilizing virtual prototypes meet the requirements set for the product development process model and development tools supporting the development of personal electronic products?

This question is answered based on the basic virtual design process model presented in Chapter 5 and the case studies presented in Chapter 7. The performance of virtual design can be examined on the general level, but in tools specific issues it is assumed that the tools used in the cases are used as a reference. The comparison of set requirements and implementations is performed for each requirement separately.

At the process model level the following can be summarized:

- *Support early validation of product concepts, versions and features:* Virtual design supports the early validation of product concepts, versions and features with virtual prototypes.

- *Support efficient communication:* Virtual design supports efficient communication with virtual prototypes integrating several design disciplines, and with a shared virtual design space integrating design information and components into a common functional and interactive product presentation.

- *Cross-domain integration support*: In addition to its basic support for multi-discipline development tasks, virtual design also recognizes the other business areas linked to product development. The practical integration to these areas depends on the customization goals. By default, at least integration to marketing and customer support is offered.

- *Supports concurrent operations:* Virtual design supports concurrent operations by enabling a parallel use of virtual prototypes in different design tasks.
- *Supports component and material reuse:* Virtual design supports the reuse of prototypes and their components. Support can be extended to documents during the customization of the process model.
- *Supports adaptivity and flexibility:* The virtual design process model itself is not tied to any tools or technologies. As a process framework, it is adaptive and flexible, and thus capable of meeting the requirement set by different organizations, and those set by possible changes in its implementation environment.
- *Distributed product development:* The basic virtual design process model does not recognize the support for distributed development as a specific requirement. Support for distributed development has to be planned on the process model implementation or customization level. For example, Cybelius Maestro supports distributed development with the Maestro Server product, and by allowing the delivery of product simulations through e-mail, shared server directories and as web presentations.
- *Utilization of modern IT technology and tools:* Virtual design is implemented by using modern IT technology and tools.

The main issues at the tool level can be summarized as follows:

- *Flexible interfaces:* Cybelius Maestro has flexible interfaces based on XML and Java technologies.
- *Support for customization:* The plug-in architecture of Maestro supports tool customization. For example, code generation can be customized to different target platforms with a target specific code generation plug-in and respective components.
- *Standard or de-facto base technology:* Maestro uses standard or de-facto standard solutions provided by XML and Java technologies.

- *Flexible architecture and technology base:* Maestro has a component and plug-in based product architecture. Flexibility is further supported by XML and Java technologies.
- *Maintaining product and process related information:* Simulation model related information is included in Maestro projects. Additional information can be linked either to the simulation project or to the prototype execution via custom made extensions.

At the prototyping support level, the following can be summarized:

- *Flexible and customized presentation of the design target and the information related to it (utilization of product and process information):* See the previous item. Different views on the Maestro project via user profiling is not yet available in Maestro. Prototype presentation executed on Maestro Simulator can be customized. There are no technical obstacles in implementing either of the previous.
- *Support for cross-engineering product presentation:* Maestro support cross-engineering product presentation by its project structure. PUI, GUI, resource and software models represent different design areas in the project. Audio properties are usually offered through software components integrating a simulation model, e.g., to the audio properties of a PC.
- *Versatile import and link capabilities from different design tools:* These capabilities can be extended in Maestro with customization. For example, an access to external feature requirement system or company wide graphical resources database can be built.
- *Versatile export and link capabilities to different design tools and IT systems:* These capabilities can be extended in Maestro with customization. The generated code used further in target specific compilers or Maestro simulation integrated to the company web page are examples of fulfilling this requirement.
- *Support of the incremental design process:* Maestro can output certain components for the product implementation. When target specific code generation plug-ins are used, a real target code can be generated directly from the Maestro models.

- *Support for heterogeneous prototyping*: Maestro supports heterogeneous prototyping through its software model. Different external objects can be integrated into Maestro simulation by way of software components. For example, a real GSM card can be used as a part of the simulation.

> 4. How can the virtual design process model utilizing virtual prototypes be taken into use and implemented in a company?

This question is not answered directly in the thesis. However, the topic is covered in coming Chapter 8.4. Practical suggestions for the implementation of virtual design can also be found in the summative case study of virtual design in the development of mobile phone UI software given in Chapter 7.3.

8.3 Contribution of the thesis

This thesis reports a research in virtual design. The thesis contributes to the research areas of engineering management, new product development and product design strategies. The contribution of the thesis for the academic community can be summarized in the following items:

- The business area of personal electronic products is introduced, studied and analyzed. Personal electronics is a relatively new research area and the research has been mainly focusing on technical innovations and their implications on users and user interactions. The area has not often been studied from the wider perspective of development and business practices and models. This thesis provides a valuable analysis of companies developing and making business with personal electronic products. The practices used and the problems faced by these companies are reported and analyzed. It was found out that many practices coming either from traditional industries or software industry cannot be applied easily as such in this highly competitive business area producing user-centred and multi-engineering products for global markets. This insight opens several new research questions and topics for researchers in the areas of CE, systems engineering and software engineering. Additionally, the description and analysis of the personal electronic business should be valuable for UCD and usability researchers as it offers them realistic and up-to-date information on the business environments they should be contributing to. The real business with its real challenges in terms of schedules, resources, competition and technical questions is usually far away from the laboratory environment.

- The virtual design process model is introduced, constructed and evaluated. Virtual design introduces a new product development and business operation framework that can support the entire product life cycle of software-focused and multi-engineering products. This kind of total product life cycle or total development cycle support of software-focused electronic products has not been reported widely in the academic community. It is important for researchers to recognize that there is still need to improve the development practices and the overall development process of these products. Compared to the more traditional areas of manufacturing industry and mechanical engineering, the development practices of software-focused and multi-engineering products have not been able to offer quite as well organized and effective development task chains and tool chains. However, the virtual design approach described in this thesis introduces several principles that can help building an effective design space or a so-called 'design-machine' approach also for these products. The proposed approach integrates many of the latest research topics of software engineering, ranging from visual software design, automatic code generation and agile methodologies to CE and UCD practices. Even though some implementations for virtual design are presented and evaluated here, the work in this area is still in the beginning. Several research questions and topics need to be solved before the principles of virtual design will be more widely applied in industry.

- The definitions and role of prototyping approaches are clarified and updated, especially from the perspective of personal electronics development and business. There is a lot of research reported on different prototyping and simulation technologies, but most of this work has been focusing on specific problems in specific application areas. This history has led to a diversity of definitions and conceptions of different prototyping approaches. For example, virtual prototyping has been defined in different ways in different design areas. Furthermore, in software engineering or UCD research, there has been hardly any research focusing on the understanding of the role of prototypes from the viewpoints of more extensive product development or business support. The mechanical engineering and manufacturing industries have been the forerunners in employing and reusing digital prototypes over several product development phases and even over business tasks. In view of this, it is surprising that research projects concentrating on software-focused products have not reached this goal, even though the software is, indeed, digital in its deepest meaning. This thesis provides valuable information on the general field of prototyping in different application areas, and in the different tasks and phases of product development. Additionally, this research emphasizes the need for further research on the various prototyping technologies and approaches, so that these could be better used for recognizing and meeting the needs related to the development of software-focused and user-centred electronic products. Finally, prototyping should not be regarded only as a solution for specific tasks or problems, but it should be approached and studied also from the perspective of the entire product life cycle.

This thesis presents applied research with a strong connection to practical industrial cases. The contribution of the thesis for practitioners and for the industrial community and can be summarized in the following items:

- Virtual design is introduced for the development of personal electronics. Virtual design defines a practical development process framework that utilizes virtual prototypes over the entire product life cycle from product conception to marketing and customer support. The possible applications of virtual design and virtual prototypes with personal electronic products are demonstrated in case studies. These show that virtual design can already be of benefit in selected design tasks and business operations. However, the full benefit of the approach is achieved only when it is applied over the entire product life cycle.
- The role of prototypes in product development and selected business tasks is recognized and emphasized. The thesis emphasizes the role of prototypes in the development of personal electronics. Prototypes and simulations are utilized in many different tasks in the case companies. Generally, their role and possibilities are not recognized in overall business operations and in the product development strategy. This leads to uncontrolled prototyping practices that further lead to a waste of effort and money as prototypes are created repeatedly in different tasks without the possibility of reusing the

prototypes and prototype components. The thesis presents practices that support an effective use of prototypes also in terms of overall business operations.

- Practical guidance and suggestions are provided for implementing virtual design in the industry. This information includes considerations for tool support (Chapters 6), experiences from real industrial cases (Chapter 7.3) and general challenges as regards implementing virtual design (Chapter 8.4).

In addition to the contributions summarized earlier for academic and industrial communities, it is worth noting that the research and innovations behind this thesis have had a major impact on the start-up of the Cybelius Maestro products and related business. The Cybelius Maestro products have been developed around the idea of virtual design, and the virtual design process model has had a central role in the business strategy and plans of Cybelius Software. Cybelius Maestro is currently used by several companies representing device manufacturers, ODMs, software platform providers and mobile operators. Even though most of the Cybelius Maestro customer references are confidential, significant companies such as Nokia Plc., Polar Electro Oy and SavaJe Technologies Inc. can be mentioned publicly. The use of Cybelius Maestro by these customers represents a strong evidence of the potential of virtual design.

8.4 Notes for practitioners

Virtual design introduces a promising approach for developing personal electronic products successfully. However, when used as a complete product development and business process model, it is not an easy approach to put into practice. Companies that are already in operation are running their existing practices and processes, and changing them is not easy. In general, the larger and older the company, the more difficult it is for it to change its operations and to adopt new practices. In practice, this means that most companies will apply virtual design starting from certain selected design tasks or domains, and then gradually expanding and enhancing its use. For new and innovative companies, the adaptation of virtual design may be easier. When these new companies recognize the dynamics of business in global high technology markets, and realize the possibilities of a new kind of business operation and way of working, they can relatively easily adjust their organization and processes already at the beginning to gain a maximum performance. This means that these businesses can also quickly and easily customize their own virtual design process to accommodate changing needs.

When evaluating the possibilities of virtual design, it is important to remember that while it can help and be of benefit already in selected limited tasks, to take full benefit of it means that is should be applied through the entire product life cycle. For example, in UI software design, if one of the design areas, such as localization does not want to use virtual prototypes coming from specification phases, it must use other tools and probably even wait for the results from the implementation (see Figure 21). This change, i.e. a break in the design flow, will immediately reduce the benefit of the approach, as costs will increase due to extra tools and the time-to-market will become longer, because localization can be done only after implementation. Based on the experience from several customer cases at Cybelius Software, it has been recognized that one of the biggest challenges for adopting virtual design in a company relates to the issue that there are no experts or managers who could see the full extent or potential of virtual design. If the benefits of virtual design are examined only from the viewpoint of a specific task or design area, the expected benefits compared to the needed investments may not appear great enough.

A third topic that has an important role in the successful implementation of virtual design is the tool support. Some of the basic ideas of virtual design can be implemented already with the currently available commercial tools, at least within certain sub-processes. However, it is clear that virtual design is still missing extensive tool support from the providers of commercial tools. Because of this, companies are forced to develop case specific, in-house modifications and extensions to commercial tools. This extra effort leads to the questions “What are the costs involved in the adoption of virtual design? Are they too high compared to the expected benefits?” This together with the tool related problems mentioned in the thesis clearly suggest that virtual design sets new demands for computer-based tools. Both of the existing tools must advance, especially in their openness and flexibility, and totally new tools must be developed in order to integrate better with domain specific tools to ensure a fluent workflow.

One issue that must be remembered in the implementation of virtual design is the process customization. As every organization and company is different, the virtual design process model must be customized to meet their specific needs. Customization always requires case specific research and planning, and it can vary considerably in terms of the scope (this department, this country, these global operations, the entire company, et cetera), the resources, schedule, and goals and vision set by the managing organization. The customization of the virtual design process model can focus only on product development, but very often it is a part of more extensive company level strategy planning, in which a company is looking for new ways and means to change its operations more effectively in general. During the planning of the implementation of virtual design, the company must understand that when adopting new practices, additional work may be needed and that benefits are to expected after a certain delay. This is a typical phenomenon, for example, in component-based development, where the building of initial components from scratch takes time, and the actual time and effort savings come only afterwards.

Besides all of the challenges faced in the implementation of virtual design, the outlook for the success of virtual design is optimistic. The tight competition in the business of personal electronics is setting increasing pressure on companies in terms of shorter time-to-market and more desirable products. This pressure will force companies to search for more effective ways of working. Virtual design represents one promising choice in meeting these strict requirements, and it is obvious that many of its basic ideas and concrete solutions will be utilized by the industry during the coming years, either under the terms of virtual design or some other corresponding design methodology. When virtual design or its main ideas in other forms start to evolve and find more extensive use in the industry, they will offer not only benefits for the companies applying them, but also new business opportunities for companies producing tools and services for virtual design implementation.

8.5 Future work and hopes

The thesis presents a basic virtual design process model framework and its implementation in the development of personal electronics. As a process and operations model, virtual design presents a wide research area, which still offers many opportunities to study. One interesting area is to research virtual design from the organization point of view. How can it be utilized in distributed product development and distributed organizations? How can it be implemented in corporate and project networks, consisting of customers and subcontractors? Specific issues related to the previous questions concern the question of how web-enabled virtual design can be supported?

There are also many other research possibilities, especially in the implementation of virtual design and virtual prototypes in different task and design areas. An example of these possibilities is provided by Mr. Mika Salmela (Salmela et al. 2001a, Salmela et al. 2001b), who studied the possibilities of virtual prototype as a front-end to PDM systems. Similar work could be done, for example, in the area of usability engineering. What kind of additional tools could be built to support usability engineering? Are there any specific requirements in virtual prototypes supporting usability engineering? Another interesting topic to study would be how virtual design and virtual prototyping could be employed in the design of new ubiquitous computing systems (cf. Tuulari 2005). A lot of additional research themes and questions are offered when virtual design and virtual prototyping tool support is considered. There are several technologies and architectures available for making new innovations in this field. Despite the fact that it was recognized that VR technologies were not mature enough for companies in the research reported here, they can be further developed and researched within the context of evolving technologies. We do not know yet if VR can become mature enough also for the prototyping of personal electronics in the future.

Having over 10 years experience in national and European research projects, and in co-operation with different research organizations, the author is concerned about the lack of research in engineering practices and engineering management regarding, especially, multi-discipline co-operation in the area of modern electronic products. It still seems that the ‘camps’ of software process practitioners, mechanical engineers and usability engineers are only focusing more and more on their own special areas, and ignoring the importance of efficient co-operation in the design areas for companies and successful business. For example, in a CE-NET, Concurrent Engineering Network of Excellence, ESPRIT project (No25946) during 1997–1999 there were only a few representatives from the software area and companies producing software-focused products, even though CE should also indeed recognize modern multi-disciplinary businesses. Additionally, for example, in Finland no strong or recognized research organizations are concentrating on this research area, even though there would be a lot to research in this area, and most probably many companies would also be highly interested in the related research.

9. Conclusion

In this thesis a virtual design process for the development of personal electronic products was introduced and analysed. Virtual design is a product development and business process framework that utilizes functional, interactive and accurate virtual prototypes over the entire product-life cycle, from concept design to marketing and customer support. The goal of the thesis was to study how virtual design and virtual prototypes could benefit the development of personal electronic products, and further, how companies could develop personal electronics more efficiently, i.e. with better customer satisfaction, reduced effort and shorter development times?

The research was started by introducing the general challenges faced in the development of modern electronic products for global markets. Next, the typical product development practices and tools used in the companies developing electronic products were analyzed. After this, the challenges and problems related to previous practices and tools were analyzed. The analyses were done based on the experiences from the electronics industry. The requirements for better development practices were also defined based on the experience and feedback from case companies. As a solution for the set requirements, the virtual design approach was introduced. A basic process model for virtual design was introduced and tool support for its implementation was discussed. Next, three design cases were presented to evaluate the potential of virtual prototypes and virtual design in the development of electronic products. One of the cases was a summative case presenting the results and experiences from real customer projects conducted at Cybelius Software with Cybelius Maestro tools.

The case studies showed that it was possible to create functional, interactive and visually accurate virtual prototypes. Additionally, they proved that virtual prototypes could be used for supporting the building of heterogeneous prototypes using existing hardware and software components and products. Similarly, it was shown that virtual prototyping supported incremental prototyping and product development, by enabling the utilization of virtual prototypes and their components directly in real product components.

The results from the design studies showed that virtual design could be used to help companies in developing personal electronics more efficiently. The benefits of the approach can be summarized as follows:

- Virtual design can improve the customer satisfaction by enabling more secure product decisions in product concepting by using realistic virtual prototypes.
- Virtual design can shorten the time-to-market by parallelizing the development tasks, marketing and customer support activities.
- Virtual design can cut costs by supporting prototype and prototype component reuse.
- Virtual design can help achieving a better product quality with component-based development and prototyping.

The thesis contributes to the research in the fields of engineering management, new product development and product design strategies. It introduces and analyses the development and the business of personal electronics as a specific research area, and contributes valuable information on this to the academic community. The special nature of personal electronics development is explored especially with regard to practices coming from traditional industries or the software industry. Furthermore, the thesis introduces virtual design as a new product development approach that aims to support the entire product life cycle of software-focused and multi-engineering products. The virtual design research reported here recognizes and highlights the need for further research on effective development practices for these products. The thesis also contributes to the research in prototyping technologies and methods. Compared to many other studies on this field, it emphasizes the role of digital prototypes and their possibilities in the development of software-focused and user-centred electronic products. Additionally, the thesis highlights the fact that prototyping should be understood and studied also from the wider perspective of process and product life cycle support.

Besides its academic contribution, the thesis serves the industrial community by introducing a virtual design method and a range of possibilities for it. The benefits of virtual design are demonstrated with design studies summarizing also experiences from real industrial projects. Based on the experience from these projects it was concluded that virtual prototyping and virtual design have potential in company use. Since virtual prototyping supports building of realistic, complete and heterogeneous prototypes, it can already be applied successfully in specific design tasks, such as concept design, usability testing and marketing. However, in order to gain full benefit of the virtual design, virtual prototypes should be used over the different design phases and other business activities, such as marketing and customer support. With this approach, a company can build the necessary tools and component repositories to make use of virtual prototypes incrementally through the entire product-life cycle, from concept design to marketing and customer support. Appropriate tools and components enable company wide reuse of prototypes and components not only within projects but also across projects.

Even though virtual prototyping introduces a promising solution for the development of personal electronics, there are several challenges in applying it within companies. One challenge is that the entire idea of virtual design is new and unknown to most companies. Additionally, putting virtual design into practice usually requires considerable changes in the practices and operations of the company. Typically, in order to be ready for virtual design or any other new working practices, a company must have a valid reason to change. Another problem is that virtual design is still missing extensive tool support from the commercial tools providers. This boils down to the requirement that in order to successfully apply virtual design, companies should either build customized tool support by using available commercial tools and self-made extensions, or wait for new kinds of commercial tools to be launched for virtual design.

Regardless of the challenges met in applying virtual design in practice, the expectations towards the approach are great. Due to tight global competition, companies are constantly seeking new, more effective working methods and practices. Virtual design offers one solution for this need. Additionally, competition puts more focus on user satisfaction and good usability of the products. This further sets requirements for the practical implementation of UCD practices and processes. Virtual design is capable of offering adequate support for early product concepting and specification phases that have an important role in UCD. The virtual prototypes created in these early phases can be used directly in usability tests and within important product decisions to demonstrate and illustrate the design choices.

Finally, it is also expected that tool vendors will be producing new tools and tool platforms that are open and expandable and as such can integrate different design tasks better into a fluent work process. This expected evolution of tools will also offer better possibilities for companies to apply virtual design in their operations. There is also a clear trend in design tools towards visual and easy modelling practices. This trend will also support easier creation of virtual prototypes for various company needs. As a final conclusion, it can be predicted that many of the basic ideas and concrete solutions introduced in virtual design will be used in companies in the future, either under the concept of virtual design or with some other similar approach.

References

3GPP. 1998. Third Generation Partnership Agreement Project. 3GPP. 12 p. [Online, cited 18th May 2005.] http://www.3gpp.org/About/3gppagre.pdf.

Agile Alliance. 2002. Agile Manifesto. http://agilemanifesto.org. [Online, cited 18th May 2005.]

Agresti, W. (ed.). 1986. New Paradigms for Software Development. Washington D.C: IEEE Computer Society Press. 295 p.

Akao, Y. 1990. Quality Function Deployment: Integrating Customer Requirements into Product Design. Cambridge, MA: Productivity Press. 369 p. ISBN 369-0-915299-41-0.

Alonso, A., Christensen, H., Baresi, L. and Heikkinen, M. 1995. IDERS: An Integrated Environment for the Development of Hard Real-Time Systems. In: Proceedings of 7th Euromicro Workshop on Real-Time Systems, Odense, Denmark, 14–16 June. Los Alamitos, CA: IEEE Computer Society Press. Pp. 4–10.

ANSI/EIA 632:1999. Processes for Engineering a System. Arlington, VA: Electronic Industries Alliance. 131 p.

Anttila, T. 1998. A Haptic Rendering System for Virtual Handheld Electronic Products. Master of Science thesis. Oulu: University of Oulu. ISBN 951-38-5232-6.

Aoyama, M. 1996. Sharing and Managing the Design Information in a Distributed Concurrent Development of Large-Scale Software Systems. In: Proceedings of IEEE COMPSAC ‘96, Seoul, Korea, 19–23 August. Los Alamitos, CA: IEEE Computer Society Press. Pp. 168–175.

Ashley, S. 1992. DARPA Initiative in Concurrent Engineering. Mechanical Engineering, Vol. 114, No. 4, pp. 54–57.

Bannon, L. J. and Schmidt, K. 1991. CSCW: Four Characters in Search of a Context. In: Baecker, R. M. (ed.). Readings in Computer-Supported Cooperative Work: Assisting Human-Human Collaboration. San Mateo, CA: Morgan Kaufmann Publishers Inc.

Battarbee, K., Säde, S., Kuutti, K., Mattelmäki, T., Keinonen, T., Teirikko, T. and Pöyhtäri, A.-M. 1999. Internet-Based Usability Testing Using 3D Virtual Prototypes. In: Bulletin of the 4th Asian Design Conference on Design Science, Nagaoka, Japan, 30–31 October. Pp. 100–109.

Beck, K. 1999. Extreme Programming Explained: Embracing Change. Reading, MA: Addison-Wesley. 190 p. ISBN 201-61641-6.

Benford, S. D. and Fahlén, L. E. 1993. A Spatial Model on Interaction in Large Virtual Environments. In: Michelis, G., Simone, C., Schmidt, Kjeld (eds.). Proceedings of the Third European Conference on Computer Supported Cooperative Work – ECSCW 93, Milan, Italy, 3–17 September. Pp. 109–124.

Bennett, G. R. 1997. The Application of Virtual Prototyping in the Development of Complex Aerospace Products. Aircraft Engineering and Aerospace Technology, Vol. 69, No. 1, pp. 19–25.

Bischofberger, W. and Pomberger, G. 1992. Prototyping-Oriented Software Development: Concepts and Tools. Berlin: Springer-Verlag.

Boehm, B. W. 1986. A Spiral Model of Software Development and Enhancement. ACM SIGSOFT Software Engineering Notes archive, Vol. 11, Issue 4 (August 1986), pp. 14–24. ISSN 0163 5948.

Bounds, G., Yorks, L., Adams, M. and Ranney, G. 1994. Beyond Total Quality Management: Toward the Emerging Paradigm. New York: McGraw-Hill. ISBN 0-07-006678-7.

Brooks, P. F. 1975. The Mythical Man-Month: Essays on Software Engineering. Reading, MA: Addison-Wesley.

Burgess, M. A. 1991. Future of Aircraft Design. Aerospace America, Vol. 29, No. 9, pp. 22–25.

Buttolo, P., Stewart, P. and Marsan, A. 2002. A Haptic Hybrid Controller for Virtual Prototyping of Vehicle Mechanisms. In: Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS 2002), Orlando, Florida, 24–25 March. IEEE Computer Society. Pp. 249–254.

BWB General Directive 250:1992. Software Development Standard for the German Federal Armed Forces, V-Model, Software Lifecycle Process Model. Koblenz, Germany: Bundesamt für Wehrtechnik und Beschattung.

Campione, M. and Walrath, K. 1998. The Java Tutorial: Object-Oriented Programming for the Internet (Java Series). 2nd Bk&cdr edition. Reading, MA: Addison-Wesley. 964 p. ISBN 0-201-1310074.

Carstensen, P., Srensen, C. and Borstrom, H. 1995. Two is Fine, Four is a Mess – Reducing Complexity of Articulation Work in Manufacturing. In: Proceedings of COOP ‘95, Antibes-Juan-les-Pins, France, 25–27 January. Sophia Antipolis: INRIA. Pp. 314–333.

Cockburn, A. 2002. Agile Software Development. Reading, MA: Addison-Wesley. 276 p. ISBN 0-201-69969-9.

Comeau, C. P. and Bryan, J. S. 1961. Headsight Television System Provides Remote Surveillance. Electronics, 10 November (34), pp. 86–90.

Connell, J. and Shafer, L. 1989. Structured Rapid Prototyping – an Evolutionary Approach to Software Development. New Jersey: Prentice-Hall. 299 p.

Cooper, R. G. 1994. Third Generation New Product Processes. Journal of Product Innovation Management, No. 11, pp. 3–14.

Cooper, R. G. 1998. Benchmarking New Product Performance: Results of the best practice study. European Management Journal, Vol. 16, No. 1, pp. 1–17.

Cybelius Software [Kerttula, M. and Salmela, M.]. 2005a. Cybelius Maestro – Introduction and Methodology Guide, v1.0. 119 p.

Cybelius Software. 2005b. Cybelius Software sold advanced software tools to Japan. http://www.cybelius.com/images/pdf/CybeliusEmbodyPressRelease20050211.pdf. [Press release. Online, cited 18th May 2005.]

Dai, W., Huang, D., Chang, C.-C. and Courtoy, M. 2003. Silicon Virtual Prototyping: the New Cockpit for Nanometer Chip Design. In: Proceedings of the ASP-DAC 2003, Design Automation Conference, Asia and South Pacific, Kitakyushu, Japan, 21–24 January. Pp. 635–639.

Dani, T. H., Fathallah, M. and Gadh, R. 1994. COVIRDS: An Architecture for a Conceptual Virtual Design System. In: Proceedings of ASME National Design Engineering Conference, Chicago, 14–16 March. New York: ASME, DE-Vol. 67, Design for Manufacturability. Pp. 19–26.

Davenport, T. H. and Prusak, L. 1998. Working Knowledge: How Organizations Manage What They Know. Boston, MA: Harvard Business School Press. 199 p. ISBN 0-87584-655-6.

Dearnley, P. A. and Mayhew, P. J. 1983. In Favour of System Prototypes and their Integration into the Systems Development Cycle. The Computer Journal, Vol. 26, No. 1, pp. 36–42.

Dionisio, J., Henrich, V., Jakob, U., Rettig, A. and Ziegler, R. 1997. Virtual Touch: Haptic Interfaces in Virtual Environments. Computer & Graphics, Vol. 21, No. 4, pp. 459–468.

Elvekrok, D. R. 1997. Concurrent Engineering in Ship Design. Journal of Ship Production, Vol. 13. No. 4, pp. 258–269.

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

Evans, B. 1988. Simultaneous Engineering. Mechanical Engineering, Vol. 110, No. 2, pp. 38–39.

FED-STD-1037C:1996. Telecommunications: Glossary of Telecommunication Terms. Arlington, VA: General Services Administration (GSA), Information Technology Service.

Fridenfalk, M. Lorentzon, U. and Bolmsjö, G. 1999. Virtual Prototyping and Experience of Modular Robotics Design based on User Involvement. In: Bühler, C. and Knops, H. (eds.). Proceedings of the 5th European Conference for the Advancement of Assistive Technology, Düsseldorf, Germany, 1–4 November. Amsterdam: IOS Publishing. Pp. 308–315.

Gabriel, R. P. 1989. Draft Report on Requirements for a Common Prototyping System, ACM Sigplan Notices, Vol. 23. No. 3, pp. 93–165.

Haug, E. J., Kuhl, J. G. and Tsai, F. F. 1993. Virtual Prototyping for Mechanical System Concurrent Engineering. In: Haug E. J. (ed.). Concurrent Engineering: Tools and Technologies for Mechanical System Design. Berlin Heidelberg: Springer-Verlag. Pp. 851–879. (NATO ASI Series F: Computer and Systems Sciences, Vol. 108.)

Heikkinen, M. 1997. A Concurrent Engineering Process for Embedded Systems Development. Academic Licentiate Thesis, University of Oulu. Espoo: VTT Technical Research Centre of Finland. 93 p. (VTT Publications 313.) ISBN 951-38-5062-5.

Hekmatpour, S. and Ince, D. 1988. Software Prototyping, Formal Methods and VDM. Reading, MA: Addison-Wesley. 222 p.

Hix, D. and Hartson, H. R. 1993. Developing User Interfaces. New York: John Wiley & Sons, Inc. 381 p.

Horvath, I. and Rusak, Z. 2001. Collaborative Shape Conceptualization in Virtual Design Environments. Communications of the ACM, Vol. 44, No. 12, pp. 59–63.

Hult, M. and Lennung, S.-Å. 1980. Towards a Definition of Action Research: A Note and Bibliography. Journal of Management Studies, May 1980, pp. 241–250.

IEEE 828:1990. IEEE Standard for Software Configuration Management Plans (ANSI). New Jersey: The Institute of Electrical and Electronics Engineers. 16 p.

IEEE 1219:1993. IEEE Standard for Software Maintenance (ANSI). New Jersey: The Institute of Electrical and Electronics Engineers. 39 p.

IEEE 1220:1998. IEEE Standard for Application and Management of the Systems Engineering Process. New York: Institute of Electrical and Electronic Engineers. 80 p. ISBN 0738115436.

IIvari, J. 1991. A Paradigmatic Analysis of Contemporary Schools of IS Development. The European Journal of Information Systems, Vol. 1, No. 4, pp. 249–272.

ISO 9000-3:1991. Quality management and quality assurance standards – Part 3: Guidelines for application of ISO9001 on the development, supply and maintenance of software. Geneva, Switzerland: ISO/IEC Copyright office. 13 p.

ISO 13407:1999. Human-centred Design Processes for Interactive Systems. Geneva, Switzerland: ISO/IEC Copyright office. 26 p.

ISO 9001:2000. Quality Management Systems – Requirements. International Standardization Office. Geneva, Switzerland: ISO/IEC Copyright office. 23 p.

ISO/DIS 9241-11:1998. Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) – Part 11: Guidance on Usability. Geneva, Switzerland: ISO/IEC Copyright office. 22 p.

ISO/IEC 12207:1995. Informational Technology – Software Life Cycle Processes. Geneva, Switzerland: ISO/IEC Copyright Office. 57 p.

ISO/IEC 14772:2004. Informational Technology – Computer Graphics and Image Processing – The Virtual Reality Modeling Language (VRML) – Part 2: External Authoring Interface (EAI). Geneva, Switzerland: ISO/IEC Copyright Office. 49 p.

Jasnoch, U. and Anderson, B. 1996. Integration Techniques for Distributed Visualization within a Virtual Prototyping Environment. In: Grinstein, G. G. and Erbacher, R. F. (eds.). Proceedings of IS&T/SPIE Conference on Visual Data Exploration and Analysis III, San Jose, CA, 31 January – 2 February. SPIE Vol. 2656. Pp. 226–237.

Johansen, R. 1988. Groupware: Computer Support for Business Teams. N.Y.: The Free Press. 205 p.

Jokela, T. 2001. Assessment of User-Centred Design Processes as a Basis for Improvement Action, An Experimental Study in Industrial Settings. Academic Dissertation. Oulu: University of Oulu. 168 p. (A374.) ISBN 951-42-6550-5.

Jokela, T. and Pirkola, J. 1999. Using Lo-Fi Prototypes in the Development of User Interfaces for Cellular Phones. In: Bullinger, H.-J. and Vossen, P. H. (eds.). Adjunct Conference Proceedings of 8th International Conference on Human-Computer Interaction (HCI International '99), Munich, Germany, 22–27 August. Munich, Germany: Frauenhofer IRB Verlag. Pp. 305–306. ISBN 3-8167-5291-8.

Järvinen, P. 1999. On Research Methods. Tampere, Finland: Opinpaja Oy. ISBN 951-97113-9-2.

Kalawsky, R. S. 1994. The Science of Virtual Reality and Virtual Environments. Reading, MA: Addison-Wesley. 405 p. ISBN 0-201-63171-7.

Kamath, R. R. and Liker, J. K. 1994. A Second Look at Japanese Product Development. Harvard Business Review, Issue: November-December, pp. 154–170.

Keinonen, T., Nieminen, M., Riihiaho, S. and Säde, S. 1996. Designing Usable Smart Products. Espoo, Finland: Helsinki University of Technology. 82 p. (TKO-C81.) ISBN 951-22-3188-3.

Kerttula, M. and Similä, J. 1996. Virtuaaliprototypointi elektroniikka- ja tietoliikennetuotteiden tuotekehityksessä ja markkinoinnissa. In: Blanko'96 seminar, Oulu, Finland, 24–25 October. 9 p. ISBN 951-42-4500-8. (In Finnish.)

Kerttula, M., Pulli, P. and Similä, J. 1996. Product Development, Marketing and Mobility Support in Virtual Enterprise. In: Proceedings of the International Conference on Concurrent Enterprise (ICE'96), Milan, Italy, 7–8 November. 12 p.

Kerttula, M., Salmela, Marko and Heikkinen, M. 1997a. Virtual Reality Prototyping – a Framework for the Development of Electronics and Telecommunication Products. In: Proceedings of the 8th IEEE International Workshop on Rapid System Prototyping, Chapel Hill, North Carolina, 24–26 June. Pp. 2–11. ISBN 0-8186-8064-4.

Kerttula, M., Salmela, Marko, Similä, J. and Kuosmanen, K. 1997b. Improving the Product Development Process with Virtual Reality Prototyping Approach. In: Pawar, K. S. (ed.). Proceedings of the 4th International Conference of Concurrent Enterprising (ICE '97), Nottingham, UK, 8–10 October. Nottingham, UK: University of Nottingham. Pp. 109–118. ISBN 0-951-9759-6X.

Kerttula, M., Tuikka, T., Similä, J. and Pulli, P. 1997c. Virtual Reality Prototyping in Development and Web-based Marketing of Future Electronics Products. In: Wojtkowski, W. G. (ed.). Systems Development Methods for the Next Century. Proceedings of the Sixth International Conference on Information Systems Development, Boise, Idaho, USA, 12–14 August. New York: Plenum Press. Pp. 529–544. ISBN 0-306-45693-1.

Kerttula, M. and Palo, J. 1999. Value Chain of Functional and Interactive Product Models in E-business. In: Bullinger, H.-J. and Vossen, P. H. (eds.). Adjunct Conference Proceedings of 8th International Conference on Human-Computer Interaction, Munich, Germany, 22–27 August. Munich: Frauenhofer IRB Verlag. Pp. 305–306. ISBN 3-8167-5291-8.

Kerttula, M., Battarbee, K., Kuutti, K., Palo, J., Pulli, P., Pärnänen, P., Salmela, Marko, Säde, S., Tokkonen, T. and Tuikka, T. 1999a. New Product Development based on Virtual Reality Prototyping – Tuotesuunnittelu virtuaalimaailmassa. Helsinki: Metalliteollisuuden Kustannus Oy. 97 p. (MET-julkaisu 13/99.) ISBN 951-817-719-8.

Kerttula, M., Kuosmanen, K. and Hof, C. 1999b. Web-enabled Enterprising – New Requirements for Web Tool. In: Wogum, N., Thoben, K.-D. and Pawar, K. S. (eds.). Proceedings of the 5th International Conference of Concurrent Enterprising (ICE '99), Hague, the Netherlands, 15–17 March. Nottingham, UK: Centre for concurrent enterprising (University of Nottingham). Pp. 327–335. ISBN 0-9519759-8-6.

Kerttula, M. and Tokkonen, T. 2001. Virtual Design of Multi-Engineering Electronics Systems. IEEE Computer, November, pp. 71–79.

Kerttula, M. and Tokkonen, T. 2002. The Total User Experience – How to make it Positive in Future Wireless Systems and Services. In: 5th WWRF Meeting, Working Group 1 – The User in the Driving Seat (Working paper), Tempe, Arizona, 7–8 March.

Ketola, P. 2002. Integrating Usability with Concurrent Engineering in Mobile Phone Development. Academic Dissertation. Tampere: University of Tampere. Department of Computer and Information Sciences. (A2002-5.) ISBN 951-44-5359-X.

Kiljander, H. 1997. User Interface Prototyping of Handportable Communication Products. Academic Licentiate Thesis. Espoo, Finland: Helsinki University of Technology. 122 p.

Kiljander, H. 2004. Evolution and Usability of Mobile Phone Interaction Styles. Academic Dissertation. Espoo: Helsinki University of Technology. 234 p. (Publications in Telecommunications Software and Multimedia. TML-A8.) ISBN 951-22-7319-5.

Kuutti, K., Tuikka, T., Kerttula, M., Pulli, P., Salmela, Marko and Lehtonen, J. 1997. Communication and Cooperation in a Virtual Design Space. In: Proceedings of the Portland International Conference on Management of Engineering and Technology (PICMET '97), Portland, Oregon, 23–31 July. P. 112. ISBN 1-8908943-00-8. [Abstract on paper proceedings, full paper on CD-ROM.]

Lantz, K. 1986. The Prototyping Methodology. New Jersey, Englewood Cliffs: Prentice-Hall. 191 p.

Lau, H. Y. K., Mak, K. L. and Lu, M. T. H. 2003. A Virtual Design Platform for Interactive Product Design and Visualization. Journal of Materials Processing Technology, Vol. 139, Issues 1–3, pp. 402–407.

Lauer, J. M. and Asher, J. W. 1988. Composition research: Empirical design. New York: Oxford University Press. 320 p.

Leppälä, K., Kerttula, M. and Tuikka, T. 2003. Virtual Design of Smart Products. Helsinki: Edita Publishing Ltd – IT Press. 262 p. ISBN 951-826-688-3.

Loch, G. K. and Kayser, H. J. 1998. On the Company-Specific Nature of Project Success Drivers in Product Development. Fountainebleau, France: INSEAD. 23 p. (INSEAD Working paper, No: 98/21/TM.)

Loh, P. K. K., Hura, G. S. and Khoon, C. C. 1999.Virtual Prototyping of Cellular Phones. Software- Practice and Experience (John Wiley & Sons, Ltd.), Vol. 29, Issue 10, pp. 897–929.

Lombardo, J. S., Mihalak, E. and Osborne, S. R. 1996. Collaborative Virtual Prototyping. Johns Hopkins APL Technical Digest, Vol. 17, No. 3, pp. 295–304.

Luciano, C., Banerjee, P. and Mehrotra, S. 2001. Collaborative Virtual Design Environments: 3D Animation of Telecollaborative Anthropomorphic Avatars. Communications of the ACM, Vol. 44, Issue 12, pp. 64–67.

March, S. T. and Smith, G. F. 1995. Design and Natural Science Research on Information Technology. Decision Support Systems, Vol. 15, Issue 4, pp. 251–266.

Maxfield, J., Fernando, T. and Dew, P. 1995. A Distributed Virtual Environment for Concurrent Engineering. In: Proceedings of Virtual Reality Annual International Symposium (VRAIS'95), Raleigh, NC, 11–15 March. Pp. 162–170.

MEDEA. 2003. The MEDEA+ Design Automation Roadmap: Design Automation Solutions for Europe. Paris: MEDEA+ Office. 4th Edition. ISBN 2-9520704-1-5. [Available in http://www.medeaplus.org/webpublic/ publications/ publ_relation_eda.htm.]

Meyer, M. H. and Lehnerd, A. P. 1997. The Power of Product Platforms. New York, NY: Free Press. 267 p. ISBN 0-684-82580-5.

Mohageg, M. F. and Wagner, A. 2000. Design Considerations for Information Appliances. In: Bergman, E. (ed.). Information Appliances and Beyond: Interaction Design for Consumer Products. San Diego, CA: Academic Press. Pp. 27–51. ISBN 1-55860-600-9.

Moore, G. A. 1999. Inside the Tornado. Oxford, UK: Capstone Publishing Ltd. 244 p. ISBN 1-900961-58-X.

Nalli, K. 1997. Virtuaaliympäristön virtuaalipuhelimen liittäminen matkapuhelimeen käyttäen TAPI- ja WinSock-rajapintoja. Engineer thesis. Oulu: The Raahe Institute of Computer Engineering, Oulu Polytechnic. (In Finnish.)

Nielsen, J. 1993. Usability Engineering. Morristown, NJ: Academic Press. 358 p. ISBN 0-12-518405-0.

Nihtilä, J. 1996. Integration Mechanisms in New Product Development. Helsinki, Finland: Finnish Academy of Technology. 162 p. (Acta Polytechnica Scandinavia, Mathematics Computing and Management in Engineering Series No. 78.)

NIST. 1994. Research and Development for the National Information Infrastructure: Technical Challenges. Gaithersburg, MD: NIST.

Noor, A. K. and Ellis, S. R. 1996. Engineering in a Virtual Environment. Aerospace America, Vol. 34, No. 7, pp. 32–37.

Noore, A. and Lawson, M. 1992. Electronic Product Design and Manufacture in a Concurrent Engineering Environment. IEEE Transactions on Consumer Electronics, Vol. 38, No. 3, pp. 666–670.

Norman, D. A. 1988. The Psychology of Everyday Things. New York, NY: Basic Books. 257 p.

Norman, D. A. 1998. The Invisible Computer: Why Good Products Can Fail, the Personal Computer is so Complex, and Information Appliances are the Solution. Cambridge, MA: MIT Press.

Norman, D. A. and Draper, S. W. (eds.). 1986. User Centered System Design: A New Perspective on Human-Computer Interaction, Hillsdale N.J.: Lawrence Earlbaum & Associates. 526 p. ISBN 0-89859-872-9.

Nunamaker, J., Chen, M. and Purdin, T. D. M. 1991. Systems Development in Information Systems Research. Journal of Management Information Systems, Vol. 5, No. 3, pp. 89–106.

Pahng, F., Bae, S. and Wallace, D. 1998. A Web-based Collaborative Design Modelling Environment. In: Proceedings of the 7th IEEE Workshop on Enabling Technologies, Infrastructure for Collaborative Enterprises (WETICE ‘98), Palo Alto, CA, 17–19 June. IEEE Computer Society. Pp. 161–167.

Pallot, M. (ed.). 1998. CE-NET White Paper on Concurrent Enterprising. CE-NET – Concurrent Engineering Network of Excellence, ESPRIT Project No 25946.

Palmer, S. and Felsing, J. 2002. A Practical Guide to Feature-Driven Development. New Jersey: Prentice Hall. ISBN 0-13-067615-2.

Park, H. and Cutkosky, M. R. 1999. Framework for Modeling Dependencies on Collaborative Engineering Processes. Research in Engineering Design, Issue 11, pp. 84–102.

Pine II, B. J. 1993. Mass Customization. Boston, MA: Harvard University Press.

Pressman, R. S. 1987. Software Engineering, A Practitioner’s Approach. Singapore: McGraw-Hill, Inc. ISBN 0-07-100232-4.

Potter, C. D. 1996. A Better Way to Crash a Car. IRIS Universe (Silicon Graphics Inc.), No. 37, pp. 44–45.

Potts, C. 1993. Software Engineering Research Revisited. IEEE Software, Vol. 19, Issue 5 (September), pp. 19–28.

Pulli, P. 1991. An Interpreter for Heterogeneous Prototypes of Embedded Software. Academic Dissertation, University of Oulu. Espoo: VTT Technical Research Centre of Finland. 46 p. + app. 76 p. (VTT Publications 79.) ISBN 951-38-3948-6.

Pulli, P. and Elmstrøm, R. 1993. IPTES: A Concurrent Engineering Approach for Real-Time Systems. Real-Time Systems, Vol. 5, Issues 2–3, pp. 139–152.

Pulli, P., Salmela, Marko and Kerttula, M. 1996. Envisioning Future Cellular Phones through Virtual Prototyping. In: OOPSLA'96 Conference on Object-Oriented Programming Systems, Languages and Applications – System Envisioning Workshop (Position paper), San Jose, CA, 6–10 October. 2 p.

Rahikkala, T. 2000. Towards Virtual Software Configuration Management. A Case Study. Academic Dissertation, University of Oulu. Espoo: VTT Technical Research Centre of Finland. 110 p. + app. 58 p. (VTT Publications 409.). ISBN 951-38-5567-8; 951-38-5568-6.
http://virtual.vtt.fi/inf/pdf/publications/2000/P409.pdf

Repo, P., Kerttula, M., Salmela, M. and Huomo, H. 2005. Virtual Product Design Case Study: The Nokia RFID Tag Reader. IEEE Pervasive Computing, Vol. 4, No. 4 (October-December), pp. 95–99.

Rettig, M. 1994. Prototyping for Tiny Fingers. Communications of the ACM, April, Vol. 37, No.4, pp. 21–27.

Rheingold, H. 1993. The Virtual Community: Homesteading on the Electronic Frontier. Reading, MA: Addison-Wesley.

Royce, W. W. 1970. Managing the Development of Large Software Systems. In: WESCON Technical Papers, v.14. Proceedings of IEEE WESCON, Los Angeles, CA, 25–28 August. Pp. 1–9.

Rundquist, A. E. 1995. Virtual Prototyping of a Synthetic Aperture Radar Processor and RASSP Benchmark. In: Proceedings of Sixth IEEE International Workshop on Rapid System Prototyping (RSP’95), Chapel Hill, NC, 7–9 June. Pp. 79–85.

Salmela, Marko and Anttila, T. 1997. A Haptic Feedback System for Virtual Prototyping of Electronics and Telecommunication Products. In: Proceedings of the Topical Workshop on Virtual Reality and Advanced Man-Machine Interfaces (ISMRC ´97), Tampere, Finland, 1–6 June. Pp. 92–97.

Salmela, Marko and Pulli, P. 1997. A Virtual Prototyping System for Electronics and Telecommunication Product Development. Online proceedings of Virtual Reality Universe ’97, Santa Clara, CA, April 2–4.

Salmela, Mika, Kerttula, M. and Kääriäinen, J. 2001a. Web-based Virtual Prototypes – an Enabling Technology for Collaborative PDM. In: Thoben, K.-D. et al. (eds.). Proceedings of the 7th International Conference on Concurrent Enterprising, ICE 2001, Bremen Germany, June 27–29. Notthingham, UK: University of Nottingham. Pp. 53–60. ISBN 0-85358-098-7.

Salmela, Mika, Kääriäinen, J. and Kerttula, M. 2001b. Case Study of Collaborative PDM for Multi-Engineering Products. In: Roy, R. and Prasad, B. (eds.). Proceedings of the Conference on Concurrent Engineering: Research and Applications – CE2001. Anaheim, US, July 28 – August 1. CETEAM. 8p. ISBN 0-9710461-0-7. (CD-ROM proceedings.)

Schaaf, J. C. and Thompson, F. L. 1997. System Concept Development with Virtual Prototyping. In: Andradbttir, S., Healy, K. J., Withers, D. H. and Nelson, B. L. (eds.). Proceedings of the 1997 Winter Simulation Conference, Atlanta, GA, 7–10 December. Pp. 941–947.

Schwaber, K. and Beedle, M. 2001. Agile Software Development with SCRUM. New Jersey: Prentice Hall. 158 p. ISBN 0-13-067634-9.

Segal, M. and Akeley, K. 1992. The OpenGL Graphics System: A Specification. Mountain View, CA: Silicon Graphics Computer Systems.

Selic, B., Gullekson, G. and Ward, P. 1994. Real-Time Object-Oriented Modeling. John Wiley & Sons Inc. 560 p.

Shackel, B. 1991. Usability – Context, Framework, Design and Evaluation. In: Shackel, B. and Richardson, S. (eds.). Human Factors for Informatics Usability. Cambridge, UK: Cambridge University Press. Pp. 21–38.

Similä, J., Pulli, P. and Kerttula, M. 1997. Marketing and Product Customisation Using Virtual Products and Internet. In: Proceedings of the International Symposium on Global Engineering Networking (GEN'97), Antwerp, Belgium, 22–24 April. Pp. 313–325. ISBN 3-931466-20-5.

Simon, H. A. 1969. The Sciences of the Artificial. 1st Ed. Cambridge, MA: MIT Press. 123 p.

Smith, M. F. 1991. Software Prototyping: Adoption, Practice, and Management. London: McGraw Hill.

Stevens, R., Brook, P., Jackson, K. and Arnold, S. 1998. Systems Engineering: Coping with Complexity. London: Prentice Hall Europe. 374 p. ISBN 0-13-095085-8.

Stytz, M. R., Banks, S. B., Johnson, T. D., Lewis, J. M. and Rothermel, S. A. 2001. The Virtual SpacePlane. IEEE Computer Graphics and Applications, Vol. 21, Issue 2, March-April, pp. 42–52.

Sutherland, I. E. 1965. The Ultimate display. In: Kalenich, W. A. (ed.). Proceedings of the IFIP Congress 1965, Vol. 2, New York, NY, 24–29 May. New York: Spartan Books. Pp. 506–508.

Säde, S. 1999. Representations of Smart Product Concepts in User Interface Design. In: Green, W. S. and Jordan, P. W. (eds.). Human Factors in Product Design, Current Practices and Future Trends. London: Taylor & Francis. Pp. 64–72. ISBN 0-7484-0829-0.

Thomson, M. W., Frelund, A. R., Pallas, M. and Miller, K. D. 1987. General Motors 2.3L Quad 4 Engine. Warrendale, PA: SAE. Series. 19 p. (SAE Technical Paper.)

Tseng, M. M. and Jiao, J. 1998. Design for Mass Customization by Developing Product Family Architecture. In: Proceedings of 1998 ASME Design Engineering Technical Conferences, 3rd Design for Manufacture Conference, Atlanta, GA, USA, 13–16 September. (DETC98/DFM-5717.) ISBN 0-7918-1953-1 (CD-ROM Proceedings).

Tseng, M. M., Jiao, J. and Su, C.-J. 1997. A framework of Virtual Design for Product Customization. In: Proceedings of the 6th International Conference on Emerging Technologies and Factory Automation Proceedings, ETFA '97, 9–12 September. Pp. 7–14.

Tuikka, T. 1994. Reliability through Cooperative Work in Software Testing. In: Matthew, K. O. L., Ben-Zion, B. and Juliff, P. (eds.). Software Quality and Productivity: Theory, practice and training. IFIP Conference Proceedings 3, Hong Kong, 5–7 December. Hong Kong: Chapman & Hall. Pp. 336–341. ISBN 0-412-62960-7.

Tuikka, T. 2002. Towards Computational Instruments for Collaborating Product Concept Designers. Academic Dissertation. Oulu, Finland: University of Oulu. (Acta Univ. Oul. A385.) ISBN 951-42-6724-9.

Tuikka, T., Kerttula, M., Salmela, Marko and Paasovaara, H. 1999. Digitaaliset tuotemallit tiedonvälittäjänä: Virtuaaliprototyypit elektroniikkatuotteiden kehityksessä – Virtual Reality Prototyping in Electronics Product Development. Prosessori magazine (special edition for IST'99), November, pp. 32–34. ISSN 0357-4121. (In Finnish, English abstracts.)

Tuulari, E. 2005. Methods and Technologies for Experimenting with Ubiquitous Computing. Academic Dissertation, University of Oulu. Espoo: VTT Technical Research Centre of Finland. 136 p. (VTT Publications 560.). ISBN 951-38-6437-5; 951-38-6438-3. http://virtual.vtt.fi/inf/pdf/publications/2005/P560.pdf.

Ulrich, K. T. and Eppinger, S. D. 1995. Product Design and Development. New York: McGraw-Hill, Inc. 289 p. ISBN 0-07-065811-0.

Ulrich, K. T. and Eppinger, S. D. 2000. Product Design and Development. Boston, McGraw-Hill, Inc. 358 p. ISBN 0-07-229647-X.

Van Maanen, J. 1979. The Fact of Fiction in Organizational Ethnography. Administrative Science Quarterly, Vol. 24, No. 4, pp. 539–550.

Valderrama, C. A., Changuel, A. and Jerraya, A. A. 1997. Virtual Prototyping for Modular and Flexible Hardware-Software Systems. Journal of Design Automation for Embedded Systems, Vol. 2, No. 3–4 (May), pp. 267–182.

VDI Design Handbook 2221:1987. Systematic Approach to the Design of Technical Systems and Products (translated by Wallace, K.). Düsseldorf, Germany: VDI-Verlag.

Vonk, R. 1990. Prototyping – the Effective Use of CASE Technology. New Jersey: Prentice-Hall. 156 p.

Wernecke, J. 1994. The Inventor Mentor – Programming Object-Oriented 3D Graphics with Open Inventor. Reading, MA: Addison-Wesley Publishing Company. 513 p. ISBN 0-201-62495-8.

Winner, R. I., Pennel, J. P, Bertland, H. E. and Slusarczuk, M. M. G. 1988. The Role of Concurrent Engineering in Weapon Systems Acquisition. Alexandria, VA: Institute for Defense Analyses. 206 p. (IDA Report R-338.)

WM-data AB and Ericsson Radio Systems AB. 2005. Delta Method. http://www.deltamethod.net. [Online, cited 18th May 2005.]

Wood, D. and Kang, K. 1992. A Classification and Bibliography of Software Prototyping. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. (Technical Report, CMU/SEI-92-TR-013, ESD-92-TR-013.)

Yeh, R. 1992. Notes on Concurrent Engineering. IEEE Transactions in Knowledge & Data Engineering, Vol. 4, No. 5, pp. 407–414.

Yin, R. K. 1994. Case Study Research: Design and Methods. 2nd Ed. Thousand Oaks, CA: Sage Publications. ISBN 0-8039-5663-0.

Zimmermann, P. 1997. Concurrent Engineering Approach to the Development of the TM6000. In: Proceedings of Annual Reliability and Maintainability Symposium, January, Philadelphia, PA, USA. New Jersey: IEEE. Pp. 13–17.

Published by

Series title, number and report code of publication

VTT Publications 615
VTT–PUBS–615

Author(s)
Kerttula, Mikko

Title

Virtual Design
A Framework for the Development of Personal Electronic Products

Abstract

Personal electronic products are electronic devices that people use for their everyday tasks. They have customizable features and offer services that are mostly implemented with software and controlled with software-based user interfaces. The development of personal electronics is a demanding task for companies. It introduces a design target that requires multi-discipline co-operation and expertise. Besides for the integration of more traditional development areas of mechanical, hardware and software engineering, the demand for customer satisfaction extends the challenge with new expertise areas that focus on good usability and an appealing design of the product. In general, more effort must be placed on the early design phases that validate the correct product decisions that are necessary for a successful business in the international markets.

This thesis introduces virtual design as a product development framework for the personal electronics industry. Virtual design is based on the effective use of realistic and functional product simulations called virtual prototypes over the different product life cycle phases. The goal of this thesis is to study how virtual design can help companies in developing personal electronics more efficiently, i.e. with better customer satisfaction, reduced effort and shorter development time.

The research work is started by analysing the general challenges that usually are met during the development of a personal electronic product. Next, the development processes and tools that are usually applied in the development of electronic and telecommunication products are studied, along with their typical limitations especially regarding the development of personal electronic products. The analysis is founded on experience from industrial cases. Real case studies are also applied in defining the requirements for better product development and virtual design ultimately aiming to fulfil these requirements.

The potential of virtual design in the development of personal electronic product is evaluated by way of three different cases. The cases consist of 2 public cases and a summative case that describes the development of mobile phone user interface software with virtual design. The cases concretize the results of virtual design in the specific design tasks and also as a wider product development process framework.

This thesis demonstrates that virtual design supported by appropriate tools can improve the performance of a company developing personal electronic products. Maximum benefit can be attained if virtual prototypes are used throughout the entire product life cycle, from concept design to marketing and customer support. It is then that the concrete advantages of the approach, such as improved communication between the design team members, reduced development time with parallelized design tasks, reduced costs based on component reuse, and better product targeting based on improved concept validation, will become even more evident.

Keywords
virtual design, virtual prototyping, virtual prototype, product development, personal electronics, information appliances, development process, concurrent engineering

ISBN
951–38–6863–X (soft back ed.)
951–38–6864–8 (URL: http://www.vtt.fi/publications/index.jsp)

Series title and ISSN	Project number
VTT Publications 1235–0621 (soft back ed.) 1455–0849 (URL: http://www.vtt.fi/publications/index.jsp)	4036

Date	Language	Pages	Price
September 2006	English	218 p.	E

Commissioned by
Tekes, VTT, Federation of Finnish Metal, Engineering and Electrotechnical Industries, Seppo Säynäjäkangas Science Foundation

Contact	Sold by
VTT Technical Research Centre of Finland, Kaitoväylä 1, P.O. Box 1100 FI-90571 OULU, Finland Phone internat. +358 20 722 111 Fax +358 20 722 2320	VTT Technical Research Centre of Finland P.O.Box 1000 FI-02044 VTT, Finland Phone internat. +358 20 722 4404 Fax +358 20 722 4374

Personal electronics provides us with devices that we use in our everyday lives. These devices are commonly regarded as personal property that we can customize for our specific needs. For manufacturing companies, these products introduce a challenging business area that highlights the skills needed in effective multi-disciplinary product development, in addition to the importance of good usability and customer satisfaction. In general, the business represents a highly dynamic international business with fast moving technology and market trends.

In this thesis, the challenges of the development of personal electronics were studied based on the experience from several industrial cases. It revealed that many of the traditional product development process models, originating from either software or systems engineering, were not optimal for this area. To solve this problem, a new process model approach called virtual design was introduced. The approach is based on effective use of realistic and functional product simulations called virtual prototypes across the different product life-cycle phases.

The potential of virtual design is demonstrated with three design cases, also presenting results and experiences from real industrial projects. The overall results of applying virtual design to the development of personal electronics are encouraging. Naturally, many practical issues still need addressing before the approach can be considered for full-scale use, but many of its main principles are already applicable as such in the industry.

The thesis provides important information on the development of multi-technology and user-centred electronics products. This information can be further utilised in research focusing on product development practices and processes, as well as simulation and prototyping technologies. For the industry, the thesis highlights some basic rules that should be considered when developing personal electronics. In particular, the understanding of the role of prototypes in product development is highlighted.

Tätä julkaisua myy	Denna publikation säljs av	This publication is available from
VTT PL 1000 02044 VTT Puh. 020 722 4404 Faksi 020 722 4374	VTT PB 1000 02044 VTT Tel. 020 722 4404 Fax 020 722 4374	VTT P.O. Box 1000 FI-02044 VTT, Finland Phone internat. +358 20 722 4404 Fax +358 20 722 4374

ISBN 951–38–6863–X (soft back ed.)
ISSN 1235–0621 (soft back ed.)

ISBN 951–38–6864–8 (URL: http://www.vtt.fi/inf/pdf/)
ISSN 1455–0849 (URL: http://www.vtt.fi/inf/pdf/)

VTT PUBLICATIONS 615   Virtual Design. A Framework for the Development of Personal Electronic Products   Mikko Kerttula