ESPOO 2007 VTT WORKING PAPERS 73

HumanICT

New Human-Centred Design Method and Virtual Environments in the Design of Vehicular Working Machine Interfaces

Kaj Helin, Teemu Evilä, Juhani Viitaniemi, Susanna Aromaa,
Pekka Kilpeläinen, Leila Rannanjärvi & Pentti Vähä
VTT Technical Research Centre of Finland

Tuomo Kujala
University of Jyväskylä

Toni Pakkanen & Roope Raisamo
University of Tampere, Department of Computer Sciences

Pekka Salmenperä & Juha Miettinen
Tampere University of Technology, Institute of Machine Design

Harshada Patel
University of Nottingham

ISBN 978-951-38-6625-9 (URL: http://www.vtt.fi/publications/index.jsp)
ISSN 1459-7683 (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 5, P.O. Box 1000, FI-02044 VTT, Finland
phone internat. +358 20 722 111, fax +358 20 722 4374

VTT, Tekniikankatu 1, PL 1300, 33101 TAMPERE
puh. vaihde 020 722 111, faksi 020 722 3365

VTT, Teknikvägen 1, PB 1300, 33101 TAMMERFORS
tel. växel 020 722 111, fax 020 722 3365

VTT Technical Research Centre of Finland
Tekniikankatu 1, P.O. Box 1300, FI-33101 TAMPERE, Finland
phone internat. +358 20 722 111, fax +358 20 722 3365

Series title, number and report code of publication

VTT Working Papers 73
VTT–WORK–73

Author(s)
Helin, Kaj, Evilä, Teemu, Viitaniemi, Juhani, Aromaa, Susanna, Kujala, Tuomo, Patel, Harshada, Pakkanen, Toni, Raisamo, Roope, Salmenperä, Pekka, Miettinen, Juha, Kilpeläinen, Pekka, Rannanjärvi, Leila & Vähä, Pentti

Title

HumanICT

New Human-Centred Design Method and Virtual Environments in the Design of Vehicular Working Machine Interfaces

Abstract

In the field of movable machines and vehicles, the user demands and the requirements for versatility of products are becoming more challenging. New methods such as human-centred interface design are needed to improve the usability, safety and ergonomics of vehicular working machines. This study aimed to satisfy these needs. The objectives focused on developing new concepts for interface technologies and implementations for movable machines, and also to develop a new Virtual Reality platform for user-centred machine design. This paper introduces a new design method for user-centred design that combines the theoretical Human-Centred Design (HCD) method with practical use of VR devices and applications. Another objective was to clarify issues from the product development process point of view in order to implement the above system and methods. However, there are still problems involved in the utilisation of Virtual Reality applications and these are discussed. The four-step method introduced in this paper has been tested in industrial design workshops.

This paper based on five different studies: (1) Virtual reality-based design environment, (2) User-Centred Design method, (3) Adaptive user interface, (4) New Product Development, and (5) Case Studies in which all the other studies where evaluated. Each section has its own objectives and results. The main results and conclusions on all the projects are at the end of the paper.

ISBN
978-951-38-6625-9 (URL: http://www.vtt.fi/publications/index.jsp)

Series title and ISSN
VTT Working Papers
1459-7683 (URL: http://www.vtt.fi/publications/index.jsp)

Project number
712

Date
June 2007

Language
English

Pages
85 p. + app. 1 p.

Name of project
HumanICT – New Human Centered Method for the Design of Mobile Work Machine User Interface

Commissioned by
Tekes, Bronto Skylift, Ruukki, Vansco Electronics, John Deere Forestry, Valtra, Velsa, PKC Group, Sandvik Mining and Construction, Aplicom, ED-Design, Muodos, VTT

Keywords
virtual reality, human-centred design, human-machine interaction, design method, user interface design

Publisher
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

Contents

1. Introduction

Careful design of user interfaces in today's machines is very important in order to intuitively and efficiently make use of the machine's functionalities. Norros et al. (2003) present a roadmap focusing on Human-Technology Interaction Research and Design, which emphasises that the implementation of information technologies has increased the availability of information and connections in people's everyday and working activities. As a consequence of this, the complexity of interactions has increased. Määttä et al. (2005) highlight that the user interfaces of modern working machines will require the handling of ever-increasing simultaneous functions, and a deep understanding of both the working processes and their utilisation contexts. The users will face difficulties with information overflow, and the diversity of different devices, applications and equipment.

This problem has not been eased by the continuous introduction of new applications and updates to existing ones. Multimodal interface technologies were expected to broaden the capacity of information technology (ICT). However, the new ways of applying ICT have simultaneously deepened the demands on usability. Moreover, questions regarding safety and operability, security and personal privacy have gained new relevance. A human-centred design approach will meet these problem situations by involving real end users early on in the design and development process.

In theory, plenty of knowledge on good practices is known to avoid most common design errors (Barnes 1968, Chaffin 1975, Rasmussen 1986, Woodson 1981, Green 1999, Vicente 1999). Designer should be aware of the issues of importance to the end users and should understand their needs and any usability problems they are experiencing. Observing real end users can highlight any problems or areas that could potentially be improved. A human-centred approach should be used to develop new Virtual Reality (VR) applications. When users are involved in the early stages of the design process, development and evaluation, it ensures there are no or fewer usability problems after implementation of the final product or process (Viitaniemi et al. 2006) However, if the product process does not have specific process phases or functions or resources to accommodate user involvement, the outcome is quite obviously garbage in garbage out (GIGO).

The company's product process should ensure there are relevant channels in place to communicate current end user problems or issues to the designer in order to improve future designs. This will enable the designer to perform his or her job more effectively in the real world or in a Virtual Environment (VE). For example, a VR/VE system would enable the designer to design more effectively, thus also improving current work processes. Designers and researchers generally agree that the use of VR/VE systems can lead to a competitive advantage by accelerating and increasing the efficiency of the product design cycle (Määttä et al. 2003).

Besides the technical problems and barriers, there may be other factors, such as usability problems and user acceptance issues, related to the use of VR/VE applications. The use of a new human-centred design method with iteration cycles for design processes is a way to address these issues (Viitaniemi et al. 2001). Following a human-centred approach to design in addition to the development of a VR/VE system in a design context could help to meet the user needs as optimally as possible.

The main goal of this study was to develop a new user-centred design concept for the cabin interfaces in movable machines and vehicles by implementing information technology and the lifecycle concept. The new user-centred design concept was expected to guide the designer through the pre-defined design specifications by utilising a new four-step design method that affords detailed instructions on each of the tasks to be carried out during a particular phase of the design process. The method also aimed to help to facilitate the effective use of VEs in different stages of the design process. The objective of this project was to develop a new human-centred design method and virtual environments in the design of vehicular working machine interfaces. This paper introduces a new design method for user-centred design that combines the theoretical Human-Centred Design (HCD) method with practical use of VR devices and applications. Another objective was to clarify issues from the product development process point of view in order to implement the above system and methods.

This paper is based on five different studies: (1) Virtual reality-based design environment, (2) User-Centred Design method, (3) Adaptive user interface, (4) New Product Development, and (5) Case Studies in which all the other studies where evaluated. Each section has its own objectives and results. The main results and conclusions on all of the projects are at the end of the paper.

2. Immersive design environment

2.1 Virtual Reality

There is no widely accepted definition for the terms VR and VE and sometimes the approach is to use them synonymously. Also, the term VR has a large number of definitions, depending on the context. One can say that the VR can be often considered a natural extension to 3D computer graphics, with advanced input and output devices. In this paper there is a difference between the terms VR and VE. VR is an artificial reality, synthetic environment that gives a person a sense of reality (Jayaram 1997). The term denotes the technology and its system elements. The term VE denotes the models built in VR and is a general impression of a vivid, though artificial, environment. The VE enables the interaction between the user and the environment (Cobb et al. 1995, Kalawsky 1993). The term VE technique denotes the equipment techniques that are used to create artificial impressions and sense perceptions in a VE.

The definition of VE can be based on Zeltzer's description of three key components of VEs (Zelzer 1990 & 1992):

- **Interactivity** (the degree of interactions with user inputs and the VE's outputs, for example the parameters or variables of an object)
- **Autonomy** (the degree of the user's and virtual object's freedom to move, manipulate, etc., in the VE)
- **Presence** (the degree of presence is a measure of the fidelity of the sensory input and output channels).

The main tasks in most VEs are object selection, object manipulation, navigation and orientation. The design of novel user interfaces for movable machines and vehicles should improve and support user interaction and use of new devices. The design of new devices and interaction techniques should exploit the use of human skills and abilities in terms of designing for the different senses and potential body movements.

There are quite a few application areas in which VEs can be utilised. Perhaps the best-known application area is the use of VR applications in training. For example, driving simulators and flight simulators have proved that the learner's ability to interact and explore in the VE without real-world consequences is an effective way of improving the learner's performance because the user can practice certain predefined events, sometimes in dangerous situations, without the fear of physical injury or machine breakdown. Stone (2001) reviews some of the important human performance results of VR technologies. VR systems also have a long tradition in visualisation applications, like architectural design. In this paper the focus is on the use of a new user-centred design method and VE applications for user interfaces for cabins.

The application fields of VR have increased during the rapid development of computer hardware and software. The use of 3D CAD (Computer Aided Design) in design processes facilitates the use of already available 3D models in VR applications. Thereby, the feasibility of VR/VE as a tool for new product design and evaluation has been recognised (Wang et al. 2006).

The development of computer hardware – e.g. calculation and software applications – has made it possible to create digital worlds; in other words, VEs. Realistic scenes, high-quality visualisation and new interaction methods have led to more realistic VEs. Computer visualisation, which visualises objects in digital mode and in 3-D with computer software and hardware, supports participatory design – e.g. ergonomic analysis (Määttä 2003, Sundin 2001). The composite use of VE applications and digital human models can offer a way to improve the user's participation and provide advanced tools for the designer to improve the results of design and analysis (Määttä 2003). Therefore, VEs can be seen as a general tool for uniting different design tools.

There are, however, several technical problems that need to be resolved in order to optimise the benefits of VR applications in industry. One major problem is converting CAD data into virtual models; this has been a barrier to the widespread adoption and real use of VEs for many years (Wilson et al. 1996). There is no typical VE technology set-up and it is possible that for each company, research company, etc., using a particular display system, different interaction devices and interfaces are used. The decision to implement certain tools may be related to cost and availability rather than usability and comfort. The problem of standardisation of interface metaphors used in VR applications has led to a situation where new devices and interaction techniques are often designed for specific applications, so users may be unable to utilise knowledge gained from previous experience. So, it is important to consider the usability of interfaces/devices early on in the design process to avoid problems later in real use. It is also important to consider whether one can use the same kinds of devices in VR applications as in real movable machines or vehicles.

2.2 Usability of VR environment

It is important to carefully consider the development or selection of easy-to-learn, efficient, usable and intuitive interface tools that are effective in completing the VE tasks. The benefits of a well-designed VE cannot be properly harnessed without attention to matching the interaction tools with the demands of the task. It is important to appreciate that the nature of a menu/interface and/or tasks within a VE will make specific demands on interaction (e.g. in the types of wrist movements required when using hand-held interaction devices). In addition, certain interaction devices may be more suitable for use in a standing position than in a seated position, and so the specific demands of the task need to be considered prior to choosing an interaction device.

VE designers may find it useful to keep a VR/VE usability questionnaire at hand to use as a guide during early prototype development in order to ensure that important usability needs are being met by the new design. An expert-evaluation tool, such as VIEW-IT (Tromp & Nichols 2003) could be used for the same purposes. The VIEW-IT (View Inspection Tool) is a rapid usefulness assessment tool for VR systems based on two human-computer interaction inspection methods: cognitive walkthrough and heuristic evaluation.

Usability tests of VEs may be conducted so that an effective technology procurement decision can be made. When making technology procurement decisions, industrial VR/VE users may consider a number of different factors, such as cost effectiveness, precision, functionality and quality. Financial constraints often impact on the final decision. It is important that the technology being proposed for use actually meets, and is acceptable for, the industrial user's application needs; otherwise, it could prove to be an unnecessary and costly acquisition. Conducting evaluation studies with existing systems at accessible sites is a good way to ensure that the correct procurement decision is made.

There are a number of important human factors issues that should be assessed during the VR/VE evaluation process:

1. The effectiveness of the system (e.g. does the VE offer the functionality required by the user; system fidelity).
2. Participant performance – can the user perform the task effectively (this could be in terms of ability to design effectively, or time taken to complete a task, number of errors made, or successfully learning how to do something).
3. Usability of the VE and associated devices (e.g. ease of use, intuitiveness of interaction, appropriate object behaviour, adequate feedback on performed actions, etc.).
4. Health and safety issues – psychological and physiological effects (e.g. sickness, stress, eye discomfort, any postural problems).
5. User attitudes towards VR/VE system; user experience/likeability.
6. Utility of the system – is it useful for the company; does it meet the needs of the company; is there a transfer of outcomes of VE use into the real world?

7. Effects of VR/VE implementation on the organisation (e.g. how will/have work processes changed with the introduction of VR; how can the use of VR be successfully integrated into the current work process; are there are any special training needs?).

The exact methods used to assess the human factors and issues outlined above will depend on company constraints (e.g. time and resources available) and also on the specific VR/VE system/application.

2.3 Development of new VEs, devices and concepts: a human-centred approach

A human-centred approach should be used to develop new VE applications. When users are involved in the early stages of the design process, development and evaluation, it ensures there are no or fewer usability problems after implementation of the final product or process. The end product also meets user requirements and expectations in terms of quality and performance.

It is important that the user company has a clear goal that will guide the development process. Specific user requirements should drive development, and the processes of identifying user needs, VE specification, technical development and evaluation should include end-user involvement. The results of user testing should feed back into the design and technical development. This cycle of iterative development will help to facilitate the design of effective, usable, user-centred VEs, interaction concepts and interaction devices.

An iterative design and evaluation process is the key to the success of the end product in human-centred design (HCD). Although it is important to take guidance on technical, task and user constraints into account, the design process must also allow flexibility and facilitate creativity. In practice, the level of detail in which this approach is followed is dependent on the goals of the specific VE application and the level of resources available for the different phases of the HCD approach.

To achieve user-needs VE development:

1. User needs must be identified, both in the context of the VE and in the target domain,
2. There must be an understanding of user characteristics and the impact of interaction with the VE on the user,
3. The task to be modelled in a VE must be matched with an appropriate VR system,

4. The impact of environmental and setting constraints should be taken into account,
5. User needs must be translated into specifications the designers and developers can use,
6. User needs must be translated into an evaluation programme to assess whether specific requirements are met by the proposed VR/VE system.

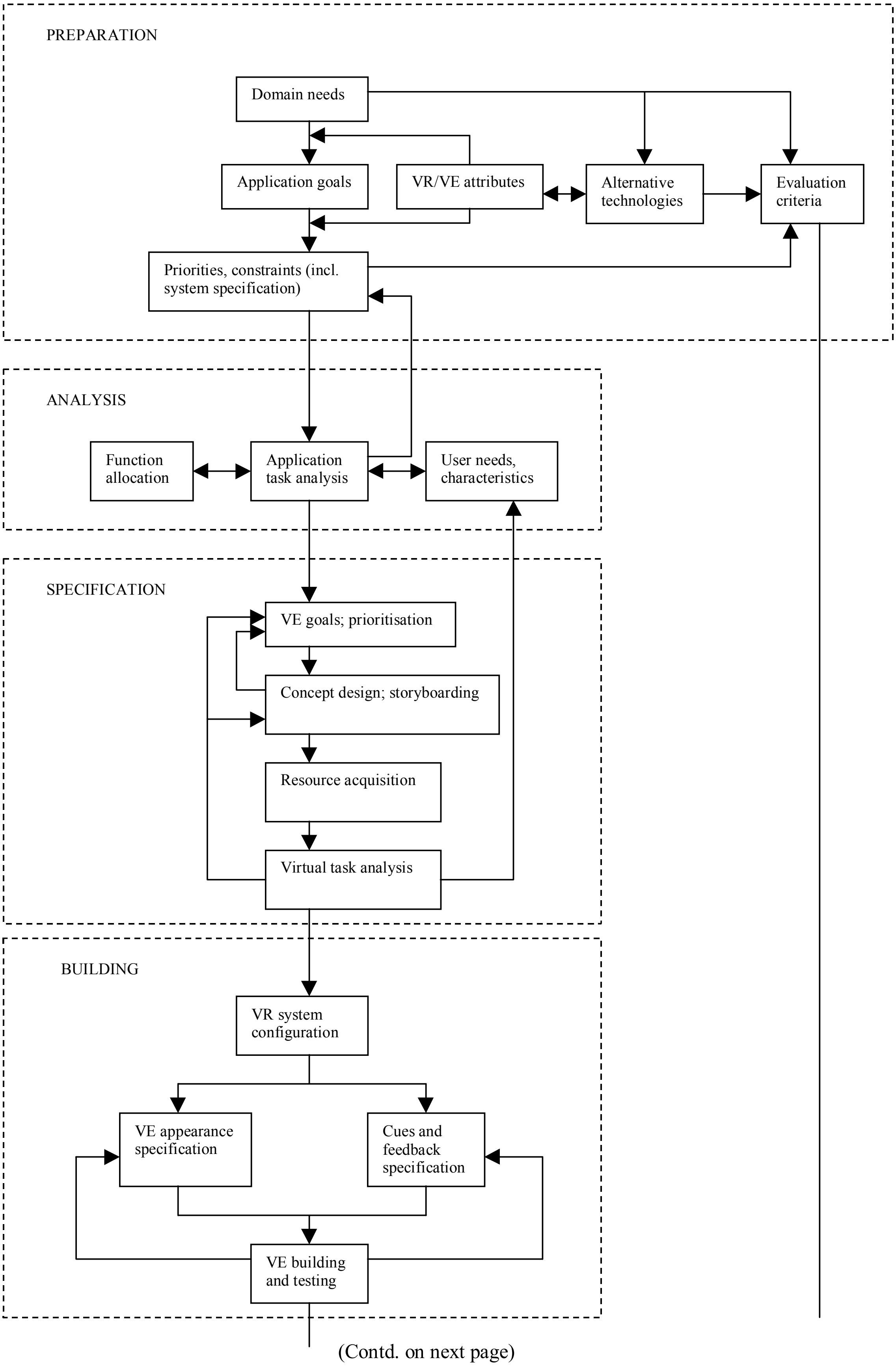
In practice, actually achieving human-centred design can sometimes be difficult. In order to develop a usable and useful VR/VE system, it is important that designers are clear about the user's specific requirements. Initially, it may be difficult to elicit useful information from the user. One way to do this is by asking the company to produce a structured document that outlines the information required in order to make procurement, design and development decisions (see below for information required from the users).

User requirements specification process for VE development:

1. Detail the context, application goals, specific objectives, priorities, constraints (e.g. technical, financial), description of task to be modelled in VR, support platforms available and user types.
2. Detail specific user needs and usability requirements.
3. Create a verifiable, unambiguous, and prioritised list of specific requirements.
4. These requirements should then be converted into clear, testable development specifications.

The user requirements document should give detailed instructions on the task to be carried out (i.e. a task analysis), thereby indicating the functionality and interaction required in the VE, and the technical capabilities required by the company. The user company should also outline their visualisation needs (e.g. in terms of image quality) and describe any specific object behaviours they require (e.g. collision detection). On the basis of this document, the user company and the system developers must decide whether VR technology can help to achieve the company's application goals. If it is agreed that VR technology can be used to meet these goals, the user company and the system developers can make decisions on the functional specifications of the VE (e.g. the level of interactivity, etc.) and the user interface and tools to be employed. Current VR technology may not be able to meet all of the company's requirements so it is important for the end user and the developers to work closely together so that there is a clear understanding of what is technically viable and the type of functionality that can actually be delivered.

The user requirements specification process could also highlight specific usability requirements. The user and technical needs that have been identified for new applications should then lead to an iterative design, development and evaluation process of new prototype VEs and devices, with end-user involvement in all stages. The user requirements specification approach outlined above was followed in the View of the Future project (IST-2000-26089) in order to develop new VR/VE applications, with successful results for the VE development approach adopted (see Figure 1 and Wilson et al. (2001) for more information on the Virtual Environment Development Structure).



(Contd. on next page)

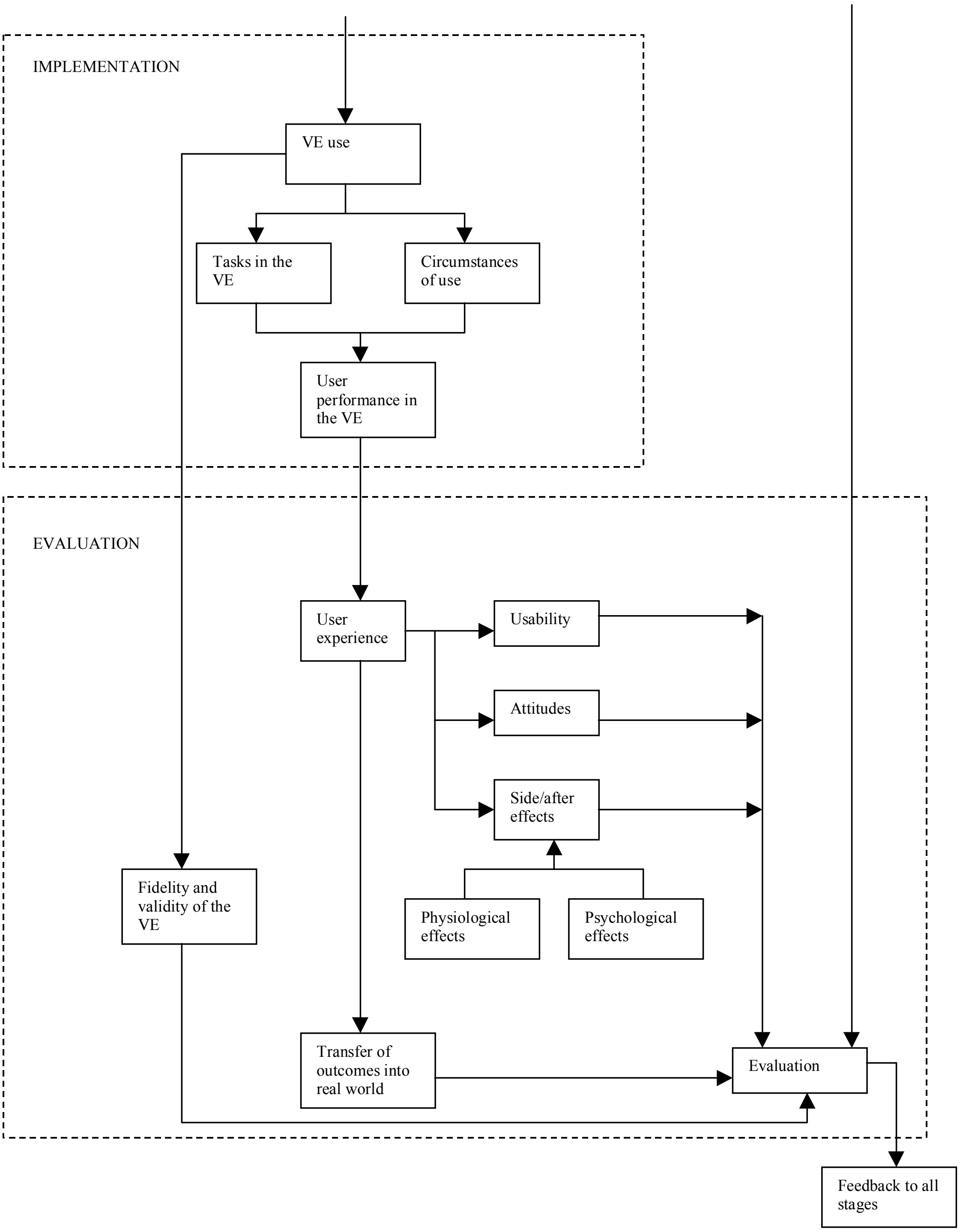


Figure 1. VEDS: Virtual Environment Development Structure (Wilson et al. 2001).

2.4 Selecting a suitable VR software

This section of the paper is for helping to choose appropriate virtual reality software for use in research and industrial applications. In Section 2.4.1 some basic issues will be taken into account and in Section 2.4.2 a simple-to-use method will be briefly introduced.

2.4.1 Some issues to consider

It is not an easy task to choose suitable software, especially for VR applications. Investing money in the wrong software can be an expensive mistake a company because software replacement can take a lot of time and effort. For example, the purchase price of software is only a one criterion that should be taken account. Other criteria can be, for example, implementation cost and annual licensing pricing. If the factors presented in the list below are taken into account, the most suitable software for fulfilling the company's and the operational environment's needs may be found. The most basic factors that should be taken into account during the process of choosing software for VR applications are:

- Software company
- Pricing
- Support and training
- Compatibility with company's other software
- Operating system
- Supported data formats
- Supported VR-related hardware
- Possibility of a demo-version for longer time testing
- The architecture of the software
- Possibility and pricing of upgrading software with new modules
- The number of software licenses in use and other references.

Software Company

The background and situation of all of the possible software manufacturers should be clarified during the process of a choosing the most suitable software. The factors to be taken into account here are:

- The financial stability of the software manufacturer
- The market position and reputation of the software manufacturer
- The attitude and resources for service and technical upgrades
- Business know-how in VR software and hardware.

The financial situation and stability of a software manufacturer is almost directly related to the recourses available for, e.g., future development of the software, customer services and technical assistance. The more money the company has, the more rapidly some certain problems like bugs will be fixed. If the market position of the software manufacturer is strong and stabilized – in other words, the software is widely accepted for use in industry and research institutes – one can be quite sure that the software has proved its ability to be versatile enough for different applications and the price-quality ratio is at an acceptable level.

The field of VR is a very specific area, where knowledge from different fields of science is implemented to form a comprehensive application. So, the software manufacture's basic know-how in those fields, e.g. human and computer sciences, is needed for good effectuation of the needs of VR software users.

Price for purchasing and annual licensing costs. The politics of the use of the software.

VR software can be quite expensive. The factors mentioned next are the key factors when forming the overall costs:

- Purchase price
- Annual licensing costs
- Indirect costs, e.g. the price of the required hardware platform
- The software vendor's policies for licensing.

Some of the available VR software solutions can offer basic license configurations at a very competitive price. One should always remember that the software must include interface methods for the use of VR hardware. Some of the most basic features always needed for VR applications can be only available with the use of some extension packages provided by the software vendor or maybe a third-party player. And, in the worst case, the extension package can be more expensive that the basic software itself. In addition, one should know the real costs of the software's technical assistance fees, and the costs and availability of training.

The operating platform needed with certain software will affect the overall costs. If the software or hardware (e.g. tracking system or stereo visualization) will not work in a normal PC, the purchase price of the computer system can multiply the overall hardware costs. In addition, these special hardware platforms, like SGI computers (www.sgi.com), are more expensive to upgrade and maintain than normal PC systems, and users of those systems have to have more specialised computer science knowledge.

The licensing policies for VR software varies a lot between different software vendors. It can be possible that one package is available from one vendor and a second required package has to be ordered from another one. Sometimes it is very time-consuming and frustrating that even the vendor is not able to give information on who can deliver a certain package and what can really be done with this one. Another example is when extending a system, it can be possible that every node (a node is a computer used in a VR simulation; the more computers there are, the more calculation power can be obtained) has to contain a full system license. In this situation, upgrading a system is very expensive. In addition, floating licenses and the price for this option should be considered.

Technical support and training

In reality, all VR-related software solutions are more or less unready packages. This is due to the fact that the markets for VR software are not big enough to employ a lot of resources in the development. This fact leads to the situation that the maturity level of VR software is not high enough to provide a really ready system for different kinds of simulations. So the end user is not able to perform all the required (and promised) simulations. It is supposed that the implementation and use of new VR software is not an easy task to handle.

The biggest problems in the implementation phase are related to the configuration of the VR hardware with the software. This is due to the complicated interfaces and communication methods with different devices and software. For handling unwanted situations, software that has simple interface methods for end users to add new hardware to the software should be chosen. Maybe there can be a graphical user-interface so the user does not to have write assembly language code. In addition, there are software solutions that support a lot of different hardware applications and are flexible and open enough for the user to write and implement new drivers and interface methods. Without working technical support provided by a software vendor / manufacturer, it is impossible to fix the deficiencies and problems, so the next factors should be taken into account during the process of purchasing new VR software:

- Support by phone, web-form and e-mail.
- Is there any vendor or market representative in Finland?
- The price and efficiency of the software support.
- Designing and implementing services for a new system (can also include the hardware needed).
- Training.
- Availability of new, upgraded versions.
- Customization for fulfilling the customer’s needs.

In most cases there are no representatives in Finland. Due to this fact there can be at least one day’s delay between the question and answer if the support is located in a different time zone (e.g. in the USA). This is not a big problem, but in some cases it can lead to unwanted situations. However, the efficiency and functionality of the support should always be tested before final decisions are made. This can be done by sending e-mails and calling the help desk. By asking simple questions, how effective the support is and whether there are enough resources to work with the customers' problems can be ascertained. In addition to asking simple questions, more complex issues that cannot be directly solved should be posed. If this non-trivial issue is handled in an acceptable time period, one can be quite sure that later, when the product is really in use, the technical support will work as promised. The more complex question can be, for example, a need for a new driver, or how to install new VR hardware (e.g. tracking system) that is not directly supported by the software.

Good training in the use of new software and hardware is an essential part of the implementation of the software. Most of the VR-related software solutions are very complicated to use because they are usually more or less unfinished entities and the user interface can be quite basic and clumsy to use. So, effective use of new software and hardware can only be done with comprehensive training provided by the software vendor or manufacturer. In addition, there should be a way to customize the software for particular needs.

Compatibility with other software and systems

If there is PDM system and CAD software already in use, one has to make sure that the chosen VR software supports the data formats used in other applications. The best solution for compatibility is to choose VR software from the same software manufacturer. This situation is not usually possible, but it seems that in future all the major CAD firms will be implementing VR capabilities in their widely used software solutions.

Operating system

The software platform, e.g. the operating system, the software uses is related to the costs of implementation and operation, and the usability issues. If the software uses general and widely used operating systems (e.g. Windows), the cost of software implementation is much more acceptable than when the software needs more specialised systems. If a need for an SGI computer system with an IRIX operating system (Unix-based system) exists, the implementation and operating costs will be much higher. In addition, these not-so-widely used operating systems are quite demanding for the users. The users have to have more knowledge of computer sciences, especially in the problem situations that usually exist with VR software and hardware. However, these systems are more optimized for visualization and calculation purposes for VR applications.

Some software packages will work with more than one operating system; they are the so-called cross-platform capable ones. However, when choosing VR software, it should be borne in mind that some software packages will work with more than one operating system, but not all the available functions are cross-platform-capable. For example, some VR software solutions do work in Windows and SGI systems, but stereoscopic visualization will only work in an SGI system.

Supported file formats

Even if the VR software has 3D modelling capabilities included, a certain need to import different file formats into the software always exists. The ability to export models may be also needed. So, there should be as many supported formats as possible. Typically, the data types that can be supported are:

- VRML, VRML 2.0
- STEP
- IGES
- 3DXML
- JPG
- DWG.

When there is a need for VR software, it should be ascertained that the software purchased will work with the software applications and data types already being used. New VR software should support a certain CAD format, if that is widely used. In addition, there is graphic modelling software, e.g. 3DSMax and Maya, which should be also supported. It must be determined whether the VR software supports native formats of CAD files.

VR hardware support

The term VR hardware includes all the basic and most sophisticated hardware tools used in VR applications. These tools include:

- Tracking systems (electromagnetic, optical, etc.)
- Control devices (mouse, 3d mouse, data gloves, etc.)
- Sound systems (speakers, headphones, etc.)
- Visualization systems (3D-displays, head-mounted displays, projector systems, etc.)
- Moving bases and their control systems (pneumatic, hydraulic, etc.).

In general, solutions for VR software applications to implement VR hardware already exist. The most common VR hardware manufacturers' solutions are usually supported.

In addition, there is usually a need to implement VR hardware solutions other than the most basic ones, so one should be sure that the software purchased will be open enough to support the user’s own hardware drivers and interface solutions.

Cluster of computers

Cluster means that more than one computer will work together to get the extra calculation power needed for VR applications. Every single computer is called a node. With the use of cluster, the simulation can be decentralized from a single computer (a master computer) to one or more simulation supporting computers (slave). Usually, only the master computer has to have the full license for VR software and the slaves can use lighter – in other words, cheaper – software licenses. One has to make sure that this really is the situation, because with some software vendors all the nodes have to have a full license, and this is very expensive.

One widely used cluster of computers is the CAVE-like visualization, where one or more computers can project stereoscopic pictures onto a single screen and, with more than one screen and one computer, quite immersive visualization systems can be made.

User Interface

The user interface is the visible interface between the user and the functions of the software. Nowadays, almost all the available software has a graphical user interface, like Windows. Although VR software manufacturers have included a graphical interface and different methods of manipulation and interaction with their software solutions, some of the solutions require a lot of basic coding with no help from the graphical interface. In some of those softwares the implementation and configuration of the VR hardware can be made with the graphical user interface. This is a good solution for people who are not keen on writing code in an assembly language. The best solution is that there should be different levels of user interface methods for different users with different levels of knowledge of computer sciences. For a basic user, the graphical interface is definitely needed, and for an advanced user, more flexible and sophisticated applications can be made with the use of an assembly language. There are software solutions for at least three different levels of interface method, the highest is the graphical user interface, then there is the so-called API level for simple and fast programming, and the last level is the lowest one for assembly language coding.

Availability of a demo version

Before the final purchasing decision, different solutions must be tested in practice. Even with simple tests, like installation procedures, importing CAD files and visualization of a simple 3D object with a stereoscopic projector can give a comprehensive view of how the software works and whether it is easy and flexible enough to be used to fulfil the needs. If the installation works perfectly and the CAD files can be imported, it can be assumed that the maturity of the chosen software is at such a level that it can be used at least for some applications. In addition, it should be ensured that the software will work with the intended VR hardware.

Concurrent with demo-version testing, the software's technical support mentioned earlier (technical support and training section) should also be ascertained.

Possibility to upgrade software

Although a basic software solution can be extensive enough to perform well in different kinds of applications, it is good to know what kinds of other applications could be made in the future. Therefore, one should find out what other packages can be included later. The packages, e.g. modules, can be:

- Digital Human Model
- Dynamics simulation module
- Interfaces to CAD software other than the one used
- Other interfaces (third-party solutions) to other software solutions (e.g. to Matlab/Simulink).

References, number of licenses worldwide

Some benchmarking should be done before the final decision is made. How widely the chosen software is really used and for what purposes should be tracked down. Some good references are usually available on the manufacturers' and distributors' Internet pages. Then, some time should be spent doing some benchmarking with companies and research institutes already using the chosen software. Scientific articles, journals and other papers should be looked through to see if there are some comments related to the use of the software. International conferences and exhibitions will help find contacts with reliable vendors and software users. It should be remembered that the software manufacturer and vendor are not neutral players, but researchers and real software users will usually give more detailed information on both positive and negative issues. If the software is widely used in industry and research institutes, it can be assumed that the software may be mature enough to perform well with different kinds of applications.

2.4.2 A Simple Method for taking some attributes into account

When all the information explained in Section 2.4.1 is collected, it has to be sorted out and a comparison made between different software solutions. This chapter presents simple, but effective guidelines for the comparison. A simple comparison chart in which where all the information is collected will be shown, and the method guidelines will be explained. This part is the so-called theoretical part of the selection of new software. After the theoretical part, the testing has to be done in practice using demo-licenses for the selected software.

Comparison criteria

Different softwares can be compared using comparison criteria. Figure 2 is a flowchart of the basic tasks that should be done during the process.

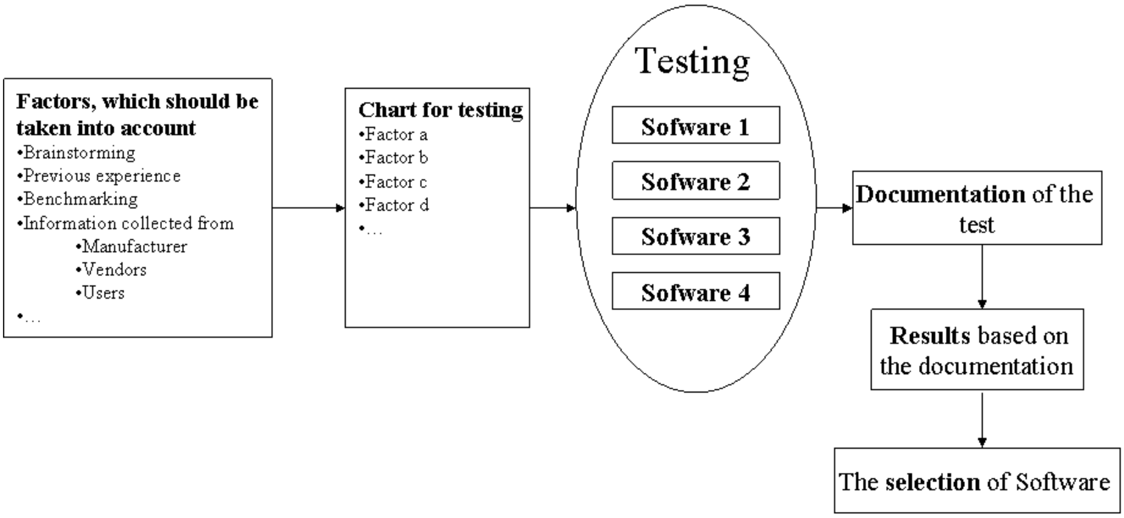


Figure 2. A flow chart for software selection.

There are three basic steps during the whole comparison process. The first is the elimination of unsuitable software solutions with simple criteria, the second is the testing and documentation of the most suitable solutions, and the last part is the selection of the best one based on the results collected during the testing period.

Comprehensive testing requires a lot of resources, so only the most suitable ones should be included in the testing. A good number of different software solutions for testing is 3 to 5. It is quite easy to eliminate some of them with simple criteria – e.g. price or support for different CAD types.

With brainstorming sessions, benchmarking, etc., the people involved in the VR software testing can collect information about the factors that are most relevant to their needs. The different factors that can be taken into account have been presented in section one. When the factors are known, a chart that takes all the relevant factors into account can be made for the testing purposes. A basic example of a comparison chart is presented in appendix A.

Practical testing

After the theoretical part, there should be 2–3 possible solutions. The practical testing itself is a quite demanding process, so more than three different solutions for testing can be too expensive and time consuming to go through. In this practical session, people who already have experience of the chosen options should perform the tests. If there is no experience involved, at least basic training provided by software vendor should be taken. Because of the complexity of the use of VR software solutions, the results collected from practical testing can be very subjective and the user maybe cannot even go through the testing if he or she does not have any training.

In this practical part, the person involved should test all the most important things promised by vendor in practice. Here is a list of questions that should be taken into account during the testing:

- Are CAD files easily used with software? What about other data types?
- Does all the VR hardware work with the software as promised?
- What is the user interface really like? Is it simple and effective to use?
- If problems do exist, how do the manuals and technical support work?
- Is it really worth all the money it will cost?
- How long does it take to model a simple, middle-sized and really large application? Is this cost-effective?

With these questions, some differences between software solutions can be found. When making the final decision, this method should show at least the advantages and disadvantages of different kinds of VR software.

2.5 Immerse design system

A prototype test environment has been designed and implemented as part of the four-step method utilising VTT’s VR laboratory facilities. The VR environment – in other words, the test environment – consists of both hardware and software applications. The test environment setup comprises the user’s interface control system (e.g. steering unit), visualisation software and hardware (e.g. 3-D projector), 2 computers and a client-server network system.

The VR software was Envision by Delmia Inc. with VR extension. Envision is a graphical 3-D simulation software for the design, evaluation and analysis of mechanical systems. The Envision software can be seen as an extension to the Igrip-robot simulation software. The software runs on an SGI Octane computer. This workstation has a single 300 MHz IP30 processor, EMXI graphics board with MXE OCO multichannel option, and 1024 Mb of main memory. The simulation models were made with the CAD software Catia V5R16. After 3-D modelling, models were imported into the Envision software and all the required functionalities (kinematics) of the simulation models were implemented in the geometry model. The Envision simulation program is mainly used for offline programming, path planning and teleoperation applications in robotics. It can also be used as a virtual prototyping tool when designing machines and devices. In this research the functions of the work machine were simulated using the Envision program. The user interface devices were connected to the simulator so that the user was able to use the functions of the virtual prototype of the work machine – see Figure 3. The multimodal user interface consisted of visual, auditory and haptic elements and it adapts itself to the current context automatically.

One PC was used to handle the interaction between the control system and the SGI computer with visualisation software. More detailed information on the configuration of this system is presented in the next Section (2.5.1).

VTT's VR laboratory facilities consist of different types of hardware related to VR techniques. For visualisation purposes, nVision's Datavisor head-mounted display (HMD) and Barco's CRT 3-D active stereo projector with 3m x 3m screen can be used. The movements of the user can be tracked with an electromagnetic tracking system. The system is Ascension's MotionStar Wireless system with 2 extended range transmitters and 16 sensors that can be attached to the user's body. The system is capable of 6 DOF tracking. 5DT's data gloves can also be used with this system.

2.5.1 Control System

The control system is made for movable machines and vehicles design and evaluation purposes. Moreover, one of the system requirements was that the system should also be generic so that it could be used during the project and later for all kinds of user interfaces for controlling virtual models. Generally, the type of user interaction in a VE is specific to a particular application and cannot readily be used in another application, at least not without making often quite difficult modifications to the system. In this project the control system was made for user interaction with simulation models of movable machines and vehicles. The control system was intended for industrial case design workshop test cases. For future research projects, the control system can be modified and extended for the emerging needs of new applications.

The system includes both control devices typical in the gaming industry, such as joysticks, steering wheels and pedals, and control devices taken from industrial machine applications, such as a harvester (Figure 3). The difference between those two genres is that the industry devices work by measuring changes in voltage increment and the gaming industry devices can be plugged into the measurement computer with, for example, a universal serial bus (USB), serial port (COM) or gaming port.

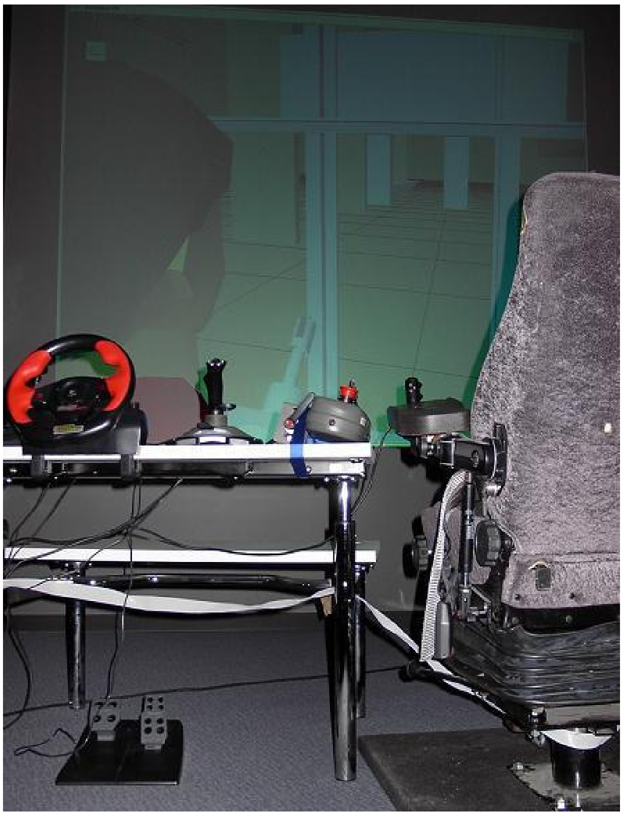


Figure 3. The control system includes interaction devices.

Input data from the industrial control devices is collected with an NI-DAQ PCI 6229 measurement card (32-analog input channels) and SCB-68 connector boxes. The voltage signal is then passed to Labview 7.1 virtual instrumentation software, which unites all signals collected from the control devices, processes them and sends them to a client computer (SGI). Figure 4 is a functional flow diagram introducing the simulation process.

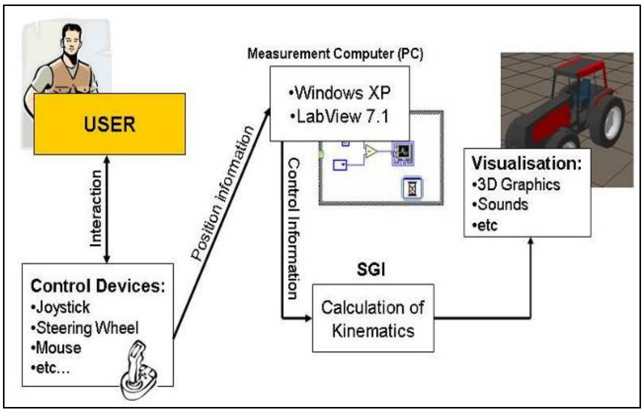


Figure 4. A flow diagram introduces the simulation system.

The interaction between two computers is handled with a Transmission Control Protocol/Internet Protocol (TCP/IP) connection. One 1.5 GHz PC is used as a server and an SGI Octane is a client. The TCP/IP protocol controls the connections and ensures that the data transfer has been done correctly and in the right order. The TCP connection also ensures that missed data packets or broken packets are registered and resent. The communication between computers should be handled in real-time constraints. Because of real-time constraints, the Nagle algorithm for flow control is disabled. The Nagle algorithm unites small packages and sends them as one bigger one. This is not a desired feature in a situation in which all the information should be delivered to all instances involved (e.g. to all nodes of a cluster of computers) as fast as possible.

The data transmission sequence (see Figure 5) is made with event-based data handling. The server (a PC with Labview application) monitors a port defined by the user (e.g. 5101) and waits for the client to create a socket and connect to a server. In practice, the socket connection is made when the client sends an ok message to a server. Then the server knows that the socket connection is made and it sends a data packet to a client. The data packet contains ordered and scaled information collected from the control devices. The data transmission sequence description is as follows:

1. Labview application monitors a port
2. Client (SGI computer with application made with Envision) sends a request for a socket connection (first ok message)
3. Labview application sends a control data packet to a client
4. Client sends an another ok message when it is ready to process another data packet.

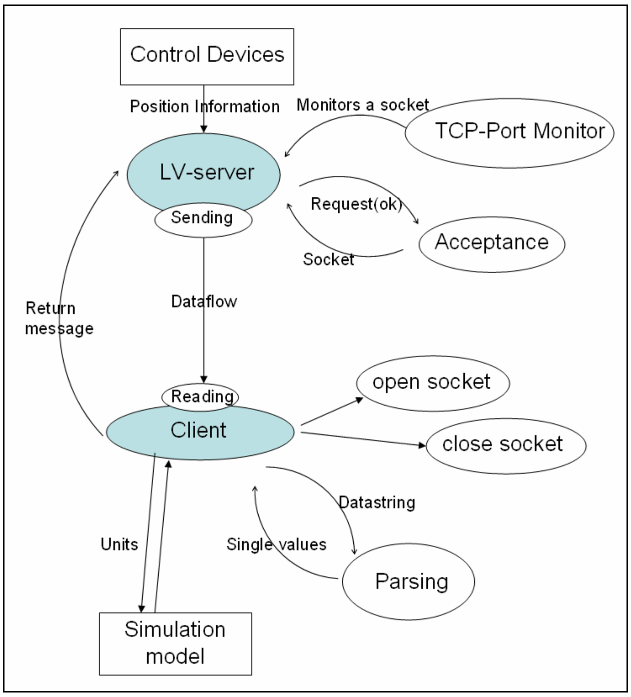


Figure 5. Flow chart of data transmission sequences.

The Envision software is graphical 3D simulation software with kinematics simulation capabilities, so all the control inputs from the interaction devices can be processed in Envision. The simulation model – in other words, the application – can be controlled with commands programmed with Envision's Graphic Simulation Language (GSL) interface. In practice, the required signals from the control devices will be connected to joints for generating simulation model movements.

3. New User-Centred Design Method

3.1 Human-Centred Design and VR

One of the main goals was to study how user and work process-related information could be more efficiently utilised in the design process for user interfaces in vehicular working machines. The purpose was to investigate how HCD as defined in the standard ISO 13407 (ISO 13407 1999) as suitable for the design of working machines. The research aimed to discover the methods and data sources in the design process that are the most crucial for a successful design from the user's point of view in this particular application area.

An additional target was to investigate how modern tools, especially VR applications, could facilitate taking the user to the centre of the design process. In the automobile industry and related research, simulators are already widely used to investigate the safety effects of in-vehicle telematic systems and, for example, the driving abilities of elderly drivers (Bullinger & Dangelmaier 2003, Lee et al. 2003, Smith 2001). According to research by Santos et al. (2005) and Engström et al. (2005), the results with simulators are found to be fairly consistent with the results obtained in real vehicles. The use of a simulator in the automobile field can be justified by ensuring the safety of the subjects, the possibility to use high-risk traffic conditions or environment, and the good access to traffic, vehicle and subject data (Bullinger & Dangelmaier 2003). These benefits of VEs could also be applicable in the design process of vehicularworking machine cockpits.

The developed user-centred design method was based on the HCD standard (ISO 13407 1999), literature review, user observation in the field, and workshops in which experts and designers discussed the proposed method.

3.1.1 Human-centred design

HCD is defined as "a multidisciplinary activity, incorporating human factors and ergonomics knowledge and techniques with the objective of enhancing effectiveness and efficiency, improving human working conditions, and counteracting possible adverse effects of use on human health, safety, and performance" (ISO 13407 1999). As such, the design process is very widely defined in the standard. There are no explicit examples of how the user should be added to the process. This leaves open a very wide range of possibilities to add the user in to the design process. It also raises the question does the mere involvement of the end users of the product in the design process guarantee a successful design?

Norman (2005) argues that HCD can even be a misleading or harmful approach. Norman states that it is wrong to obey the users excessively. Although some user opinions should be listened to attentively, the designer should ignore the comments if they are against the better knowledge of activities the technology is designed for. Norman's point is to replace the term human-centred with activity-centred. In this way, instead of focusing on the static qualities of the technology and the requirements for different users, he emphasizes the importance of considering what activities the technology is designed for. However, giving human activities a high priority in the design process can be argued to be a form of HCD. This approach has a close resemblance to what is defined by Saariluoma (2004) as explanatory design – instead of testing the usability of each interface prototype in the design process, we actually have the appropriate psychologically well-grounded design principles for the interaction design before the first prototype. In addition, explanatory design knowledge aims to provide information on people's activities, goals and behaviour in the particular context of use.

It can be argued that the success of a design can depend on which particular methods of HCD (customer feedback, opinion polls, task analysis, usability testing, participatory design, etc.) are emphasized in the design process (Norman 2005). The importance of a data source may well depend on the type of product under construction. For example, in video game design a higher priority should be given to users' subjective opinions and experience than in the design of a highly regulated and safety critical area, such as a nuclear plant. In the context of vehicular working environments, a high priority should naturally be given to the work processes and to reducing their demands on the user.

The importance of considering ergonomics, human factors and usability in design

Ergonomics (or Human Factors Engineering) is defined as "an applied science concerned with the characteristics of people that need to be considered in designing things that they use in order that people and things will interact most effectively and safely" (Pease 2002). The importance of ergonomics and human factors has been acknowledged, especially in aircraft cockpit design, since World War II (Kantowitz & Sorkin 1983, Reason 1990). Also, usability, as defined in the ISO standard 9241-11 (ISO 9241-11 1998) as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use", has gained increasing attention, particularly in the human computer interaction discipline since the revolution of home computers. As is evident in the definition of usability, the term can be expanded to relate to anything that can be considered technology or an artefact (Nielsen 1993, Norman 2002). In recent years, ergonomics and usability-related issues have also been gaining increasingly more attention among working machine manufacturers.

Poor usability or ergonomics of a user interface in a vehicular working machine may lead to poor work efficiency, low work satisfaction, or even endanger the occupational health of an employee. In a worst-case scenario, insufficient consideration of human factors in design may lead to fatal working accidents (Reason 1990).

An important research area to consider when dealing with vehicles, is the demand for attention the controlling of movements places on the user. In a vehicular working machine these demands can include the lateral and longitudinal control of the vehicle and the control of tools, such as hydraulic arms. The control of these movements should be simulated when testing the usability of in-vehicle systems that are meant to be used while the vehicle is moving (Burnett et al. 2004). The attentional capacity of humans is limited, and when it is exceeded, reaction times to sudden, unexpected events will be prolonged (for a review of the psychological evidence, see Cowan (2000)). This can cause severe working accidents. This is why the demands for the user's attention to, for example, in-vehicle displays should be addressed (Wierwille 1993). The increasing use of information and communication technologies in working machines underlines this issue because the user interfaces of these systems usually place additional load on the cognitive processes of the user.

3.1.2 Design Methods

There are a variety of ways in which the usability, ergonomics and human factors can be considered while the design process proceeds. Methods giving high priority to the needs of the user include:

- Task analysis (Hackos & Redish 1998)
- Cognitive work analysis (Vicente 1999)
- Contextual design (Beyer & Holtzblatt 1999)
- Usability testing (Nielsen 1993)
- Heuristic usability evaluations (Nielsen 1993)
- Ergonomics analyses (e.g. Karhu et al. 1977)
- Participatory design (Muller and Kuhn 1993).

All of the aforementioned user-centred methods can be valuable in taking the user to the core of the design. However, the utilization of all of these in a design project can be a very time-consuming and resource demanding process. Also, a particular method may be more suitable in the design of a particular product than with another. In other words, the value, applicability and priority of the methods should be made explicit for the particular area of application. When vehicular working machines become increasingly networked to larger information systems, approaches targeted at the design of complex socio-technical systems, such as cognitive work analysis and contextual design, will presumably become more relevant.

Vicente (1999) suggests that the work the system is designed for should be modelled in a formative form. By this he means that the system is described through its intrinsic constraints. These intrinsic constraints include the constraints and possibilities the work sets for the system. These constraints are the only obstacles in the way of redesigning the work and the working processes to be more efficient. According to Savioja (2003), the basic idea of this approach is to take these intrinsic constraints as the basis for the design so that the tasks performed with the system can be designed rationally in the sense that they only include the meaningful and necessary activities for achieving the goals of the work process. Consequently, these constraints – for example, the movements of the machine – should be modelled adequately in the descriptions of the work processes and also modelled in possible VR testing of design solutions.

A design method in which VR applications could be very helpful for improving the quality of the working environments is participatory design. Participatory design "spans a rich diversity of theories, practices, analyses, and actions, with the goal of working directly with users (and other stakeholders) in the design of social systems" (Muller and Kuhn 1993).

3.1.3 Participatory Design

Participatory design is a procedure in which the end users of a production process or machine have the opportunity to take part in the design process and can influence the design and development of a new product or process. In participatory design some or all of the workers who will work, for example, at a forthcoming plant take part in a number of design sessions during the different design phases (Mumford 1989).

According to Hirchheim (1989), the advantages of participation are:

1. Better result of design: As the users themselves can influence their working environment.
2. Commitment: Users will feel more ownership of the new system and implementation will be easier as they have contributed to the design.
3. Work satisfaction: Operators will be more satisfied with their job when they have influenced their working environment.
4. Training: Participation constitutes effective training for the operation of the new system, and users will generally have a greater knowledge of the work and processes.

5. Efficiency: Increased knowledge together with the new tools will lead to more effective production.
6. Participation as a value in itself: Participation is a feature of democracy and is, as a form of it, a value itself.

Participation is a logical result of the pressures to change and implement new technology. The design and development of complex systems will have more success with participatory design than without because, for example, by implementing participatory design, it will be easier to control unexpected situations (Kallela 1996). A participatory design method was used in industrial workshops in the development process of the new user-centred design method. The industrial workshops included designers from moveable working machine companies, human factors experts and researchers.

3.2 New User-Centred Design Method – The Four-Step Method

Our purpose was to clarify and specify a successful user-centred design method for vehicular working machines and, particularly, for their user interface design. We utilised the ISO 13407 HCD processes for interactive systems standard (ISO 13407 1999) as the framework for our method. The purpose was also to evaluate the applicability of VR in the different steps of the design process.

The ISO 13407 standard (ISO 13407 1999) states four activities that a user-centred design project must include:

- Understand and specify the context of use.
- Specify the user and organisational requirements.
- Produce design solutions.
- Evaluate the designs against the requirements.

On the grounds of these activities, we developed a four-step framework method for the design of user interfaces in vehicular working machines (see Figure 6), utilising VR in completing the steps. In

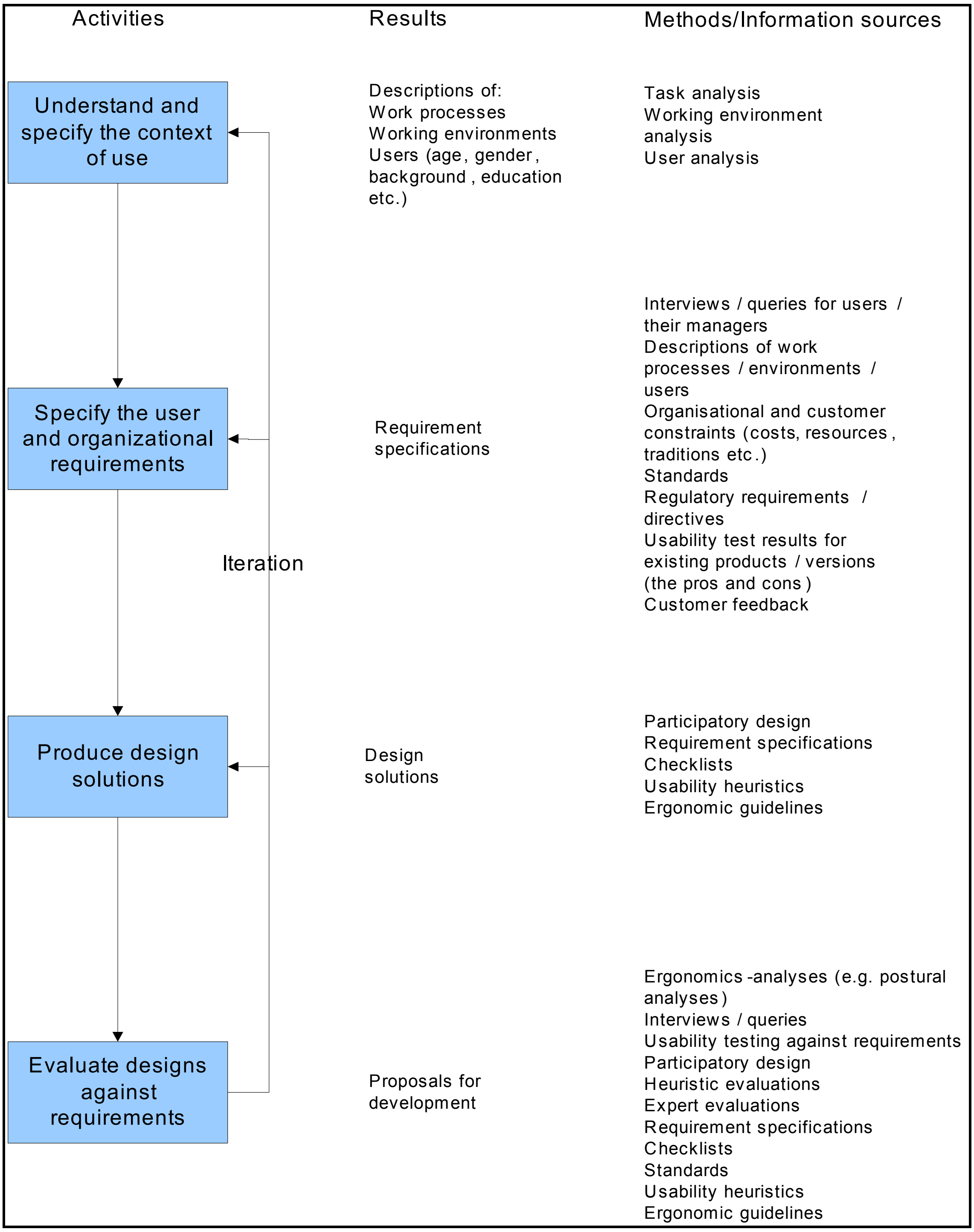
Figure 6 the activities (steps) of the design process are described in the boxes to the left, the expected results of the activities are listed in the middle column, and possible methods and information sources to help in completing a step successfully are listed in the right hand column.

Information related to work processes was given a higher priority in the design than the subjective opinions of the users. In another words, task analysis (Hackos & Redish

1998) was given the highest priority in collecting data to support design solutions. Describing the goals in a work process is a way to gather requirements and may provide an insight for a new and more efficient way to implement the interaction between the user and the machine. Customer feedback is a valuable source of information for the requirement specifications, but, in line with Norman (2005), the descriptions of activities in the work processes were regarded as the most valuable information source for the requirement specifications. In addition, the formative constraints of working environments and processes (Vicente 1999) were given a high priority in the work process descriptions.

User-centred design can be a process that demands a lot of resources. VR solutions can be utilised efficiently in the task analysis itself and also in testing alternative design solutions against the requirements the task analysis produces. Implementing a real concrete prototype of each design solution for testing how accurately it meets the requirements would be inefficient cost-wise and very time-consuming. Easily created VE models could provide a more efficient way to test the solutions against the requirements and to make different adjustments and quickly get feedback from the users , for example in participatory design sessions.

The activities of the method are described in detail below. Iteration between the steps is an essential part of the method. This means that it is possible and even desirable for the designers to make changes based on the results of earlier steps. For example, in an evaluation process the designer may discover something crucial about the working habits of the users when testing solutions with actual users. In the long run the number of iterations in the design process of a certain product type should decrease through the accumulation of explicit design knowledge. This means a more efficient production of design solutions and a decreased need for usability evaluations (Saariluoma 2004). However, the new requirements appearing in the field together with the development of technology will ensure the emergence of new design problems, and the evaluation of new solutions to these problems will also be necessary in the future.



Activities	Results	Methods/Information sources
Understand and specify the context of use	Descriptions of: Work processes Working environments Users (age, gender, background, education etc.)	Task analysis Working environment analysis User analysis
Specify the user and organizational requirements	Requirement specifications	Interviews / queries for users / their managers Descriptions of work processes / environments / users Organisational and customer constraints (costs, resources, traditions etc.) Standards Regulatory requirements / directives Usability test results for existing products / versions (the pros and cons) Customer feedback
Produce design solutions	Design solutions	Participatory design Requirement specifications Checklists Usability heuristics Ergonomic guidelines
Evaluate designs against requirements	Proposals for development	Ergonomics-analyses (e.g. postural analyses) Interviews / queries Usability testing against requirements Participatory design Heuristic evaluations Expert evaluations Requirement specifications Checklists Standards Usability heuristics Ergonomic guidelines

Figure 6. The user-centred design method for the design of user interfaces of a vehicular working machine.

3.2.1 Step 1: Understand and Specify the Context of Use

In the beginning, the following important aspects should be described in a formal form:

- The tasks the users will perform (work processes – goals, intrinsic constraints, activities and typical behaviours).
- The environment in which users will use the system (important for the development of the VR test environment and to find the significant constraints in the environment to consider in the design, e.g. lighting and possible unexpected events).
- The significant characteristics of the intended users (important for test user selection and to achieve design for all, e.g. adjustability of working positions and visibility).

3.2.2 Step 2: Specify the User and Organisational Requirements

After gaining a sufficient understanding of the context of use, the designer should gather the user and organisational requirements in relation to the descriptions created in the previous step. The VR tools can be helpful in gathering the requirements with the design team and especially in planning the testing of the measurable design goals. The requirement specification should:

- Provide a clear statement of the HCD goals.
- Identify the range of relevant users and other personnel (e.g. managers and experts) required to participate in the design for the goals to be achieved.
- Set appropriate priorities for the different requirements.
- Provide measurable benchmarks against which emerging designs can be tested.
- Include functional, usability, organisational, statutory, legislative and ergonomic requirements.
- Be adequately documented.

3.2.3 Step 3: Produce Design Solutions

Producing design solutions consists of:

- Developing outline design proposals with multi-disciplinary input, including users, usability and ergonomics experts, designers and managers.
- Making the design solutions more concrete using VR simulations, mock-ups, prototypes, etc.

3.2.4 Step 4: Evaluate Designs Against Requirements

This is an essential step that assesses whether the human-centred objectives have been met and provides feedback that can be used to improve the design and specify the design knowledge created in the previous steps. A variety of suitable evaluation methods exist, varying in their formality, rigour and user involvement, including heuristic and expert evaluations, usability testing against the measurable requirements, and participatory design. VR can provide a valuable tool for this step to comply with the financial and time constraints of the design project by accelerating the prototyping stage.

It is also important to check the coherence of the requirement specifications. This means that, for example, the user requirement specifications and technical requirement specifications are not contradictory. This evaluation is very important, especially with highly complex systems. If the requirements diverge, it is necessary to revise step 2. The evaluation step consists of:

- Showing the design solutions to users and/or experts and allowing them to perform/simulate tasks (participatory design, usability testing, expert evaluations, experiments).
- Iterating the whole process until the design objectives are met – going back to the first steps if new knowledge about the context of use and user requirements is gained, or a contradiction in the requirements is found.

4. Adaptive user interface

The interface between the human and the machine is crucial for the performance of the human-machine system. Development of mechatronic machines has been fast on the level of automation of machine operations but relatively slow in the area of user interfaces. User interfaces are still constructed on the traditional idea of commanding and monitoring single operations rather than controlling task executions. Thus more intelligent solutions to the human-machine interface (HMI) issue are required. The key requirement will be that HMI will come up with the required information at the right time in the proper way, and an example could be an adaptive HMI that filters the information presentation according to the situational requirements.

Context is any information that can be used to characterise the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application itself (Dey 2000). Context awareness is knowledge of the environment, location, situation, user, time and current task. Context awareness can be exploited in selecting the application or information, adjusting communication and adapting the user interface according to the current context (Schilit et al. 1994, Schmidt et al. 1999). Context awareness has recently become a hot topic in the areas of pervasive computing and human-machine interaction. Meanwhile, mobile communication has become commonplace, in some areas practically ubiquitous; it is becoming very attractive to offer the users of mobile devices context-related information and a number of papers have been published on context recognition for handheld devices (Mäntyjärvi 2003).

Typically, the user interfaces used in work machines do not possess any kind of context awareness or adaptability to different users or use cases. Traditionally, control has been performed by the 'button technique' – that is, by switches, levers and pedals. The advances in control electronics have brought different graphical user interfaces to show or visualise readings or accept the driver's choices (parameters or operation modes). The work machine's control system has in general been designed for an 'average' user using his/her hands or legs for control purposes. Only physical characteristics like the length of the driver have allowed adjustments. Controlling the work machine could become remarkably more functional and efficient by taking cognitive characteristics and the operational situation into consideration. Along with automation, more sensors are being added as the recognition of several aspects of the work machine's context is becoming possible.

4.1 Context-Aware Multimodal User Interfaces

Communication between the operator and the work machine control system takes place via the user interface, supporting the machine operations and offering easy and straightforward operation of the machine. The UI of a typical work machine consists of control equipment intended for driving the vehicle (brake, accelerator, clutch, steering wheel and gear stick), control equipment for the machine's tools (joysticks and switches), the necessary ancillary system (instrument cluster, indicator lights, light switches, etc.), task performance-related information and automation systems (data acquisition systems, vehicle terminals), and comfort and entertainment controls (air conditioning, radio and CD-player, and mobile phone). Thus several tens of operational switches are required for implementing the basic machine operations. In order to reduce the number of operational switches, text and number-based, or graphic multi-operational displays or controls, have been taken into use for performing different operations with the aid of a menu bar.

By context definition in work machine applications we mean the definition of the operational state of the machine with the aid of the measurement data. Thus a prerequisite for context recognition is the availability of sensors in the targets to be measured, e.g. in the machine. An adaptive user interface is context-aware, meaning that it adapts to varying situations during the operation. The situations it can adapt to may be, the ongoing task, the user, or changes in the environment and the operational state of the work machine (see Figure 7). Before the user interface can adapt to the task it has to be able to recognise the task and its phases, thus a prerequisite for context recognition is that the tasks are phased and modelled, and then the machine is able to recognise the task sequence –i.e. the context.

4.1.1 User

An adaptive user interface can adapt to the user (identification), the user's role (a member of a user group), the status of the user and or the use pattern (way of action). An intelligent UI can be personalised, if the user can be recognised and the user profile has been created. For all IT workers the most familiar user recognition practice is a userID connected to a password. Most of us would like to forget our various system – userID – password triples and use a more convenient embedded identification method. The biometric user identification method can be based on fingerprint, iris, face, hand geometry, voice, gait or hand-written signature recognition (Bolle et al. 2004, Jain & Ratha 2005). Each of these methods indicates the creation of a model before a sample can be verified and a user recognised. Biometric methods are not the focus of this paper, but they are a precondition to any personalization of the user interface.

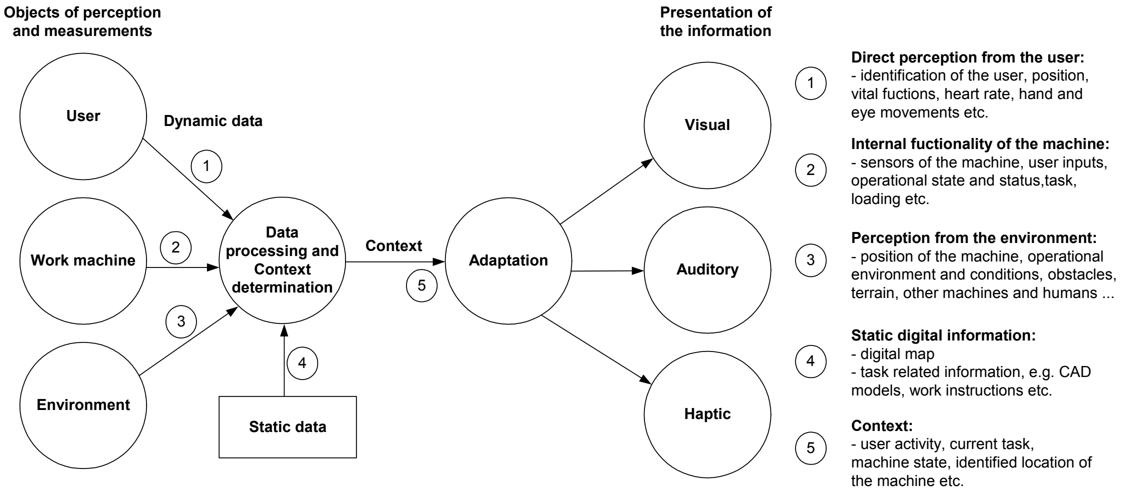

Objects of perception and measurements
User
Dynamic data
1
Work machine
2
Data processing and Context determination
Context
5
Adaptation
3
Environment
4
Static data
Presentation of the information
Visual
Auditory
Haptic
1
Direct perception from the user:
- identification of the user, position, vital fuctions, heart rate, hand and eye movements etc.
2
Internal fuctionality of the machine:
- sensors of the machine, user inputs, operational state and status,task, loading etc.
3
Perception from the environment:
- position of the machine, operational environment and conditions, obstacles, terrain, other machines and humans ...
4
Static digital information:
- digital map
- task related information, e.g. CAD models, work instructions etc.
5
Context:
- user activity, current task, machine state, identified location of the machine etc.

Figure 7. Context awareness, adaptability and multimodality in work machines.

The user profile may include attributes like:

- user group, indicating the user's level of experience.
- pattern of use, for example aggressive or sensitive control pattern.
- personal information processing style, indicating the form of feedback the user prefers.
- user role, also indicating the access for various operations. For example, the service man has to be able to carry out repair and maintenance tasks.
- physical measurements of user; after user identification, the seat, mirrors, control devices and fare collection equipment shift to the desired position.
- user status varies during the day and, depending on the current situation, workload and motivation.
- position of user varies. For example, the user may step out of the cabin to mount ancillary equipment on the machine.

In addition to the attributes mentioned above, the traditional human computer interface (HCI) research has recognised an endless list of personalization possibilities, like left / right handedness, language, culture, user preferences for radio channel, font size and type, display colour theme, etc.

4.1.2 Work machine

There are usually many sensors and electronic systems already installed in modern work machines that can be used to determine the context of the machine, and additional sensors can be attached for context determination purposes. Examples of adaptive features related to the context of the machine are:

- adaptation to the change in configuration of the machine; ancillary equipment can be mounted on the machine and the user interface of the machine has to adapt to the change. This is the usual situation with farm tractors.
- adaptation to the motion state of the machine (driving speed and direction, acceleration); only information related to the driving task is shown while the machine is being driven, and, for safety reasons, some functions of the user interface can be disabled.
- adaptation to the loading of the machine; the forces directed to the machine's tool or the mass of the load the machine is carrying can be measured and the user can be warned not to overload the machine.

Useful information for context determination purposes can be obtained from the motor controller, transmission and hydraulics, as well as joysticks and other user interface devices.

4.1.3 Environment

Adaptation to changes in the environment could trigger some automatic operations or semi-automatic sequences, depending on the situation. Critical changes can stop the current operation or try to correct the situation. This is always a trade-off between safety and smooth operation, and the decisions have to be well defined. Less critical and more pleasant adaptation features might be:

- switch the operation mode (driving off-road, driving on a road, working). Each operation mode may have automatic operations.
- adaptation to environmental restrictions, like speed limit, driving line, free work space – including, e.g., collision detection or even prevention.
- selecting the optimal way to provide feedback. For example, not to provide voice feedback in a noisy situation and reduce or add display contrast.
- providing positional information. This means, for example, navigation instructions either beforehand and also after digression.
- preventing error situations or faulty postures. This includes adjusting the posture of the cabin to an upright position on a slope or on any rough terrain.

4.1.4 Task

Depending on the type of work machine, it is usually used for a certain task or it is a general-purpose machine used for many different tasks. There can be a need to adapt the user interface to better match the requirements of the current task, especially if the machine is used for many different tasks. The user can select an adequate operation mode for that task or, if the machine has enough information on the different tasks, it can recognise the task automatically. Different languages have been developed to define work tasks for robots and automated machines, e.g. RoboML and ILMR (Intermediate Language for Mobile Robots) (Kauppi 2003).

The work machine's tasks can also be related to the location. The machine can be positioned using a GPS system or an ATS tachometer Based on the location data, the machine can interpret given digital work instructions and tell the user what to do in a certain location or automatically perform tasks. Examples of these kinds of systems are work machines used for spreading materials in road construction, e.g. road grader, excavator, bull dozer, asphalt paver. A 3D model of the surface can be given to the work machine and the machine is controlled automatically or instructions are given to the user, who controls the machine manually (Kilpeläinen et al. 2004a, Heikkilä & Jaakkola 2002). A soil stabiliser is used in the rehabilitation of old road structures, and various parameters, e.g. the stabilising agent mixture and cutting depth, can be controlled automatically according to the rehabilitation design file (Kilpeläinen et al. 2004b). The same kind of technology is used in precision farming, e.g. in seeding, fertilisation and spraying plants with protective agents. In some fields standard formats have been developed for describing the tasks of the work machines. IREDES is the unified language for Rock Excavation Machinery, OSYRIS stands for Open System for Road Information Support (Ligier et al. 2001) and LandXML is an industry-driven, open XML data exchange standard that provides interoperability in more than 40 software applications serving, e.g., the civil engineering industries.

When the task is modelled and the sequence automatically recognised in real time, the user interface (of a machine control system) may assist the user by:

- proposing the next operation (and waiting for the user acceptance, selection of parameters or another operation).
- semi-automatic operations; when the user has selected an object the work machine identifies the object and handles tasks related to the object automatically (e.g. automatic harvesting of trees).
- selecting the next work object; providing the necessary instructions (navigation plan, service manual, work description or case-sensitive recommendations).

- providing task-sensitive information – i.e. displaying the advisory tool information in the main display or pointing out a navigation error with corrective operations.
- providing (multimodal) warnings when the work object is out of range or the machine is overloaded.
- saving the implemented work into the quality system.

4.2 Application Of The Context-Aware Systems In Work Machines

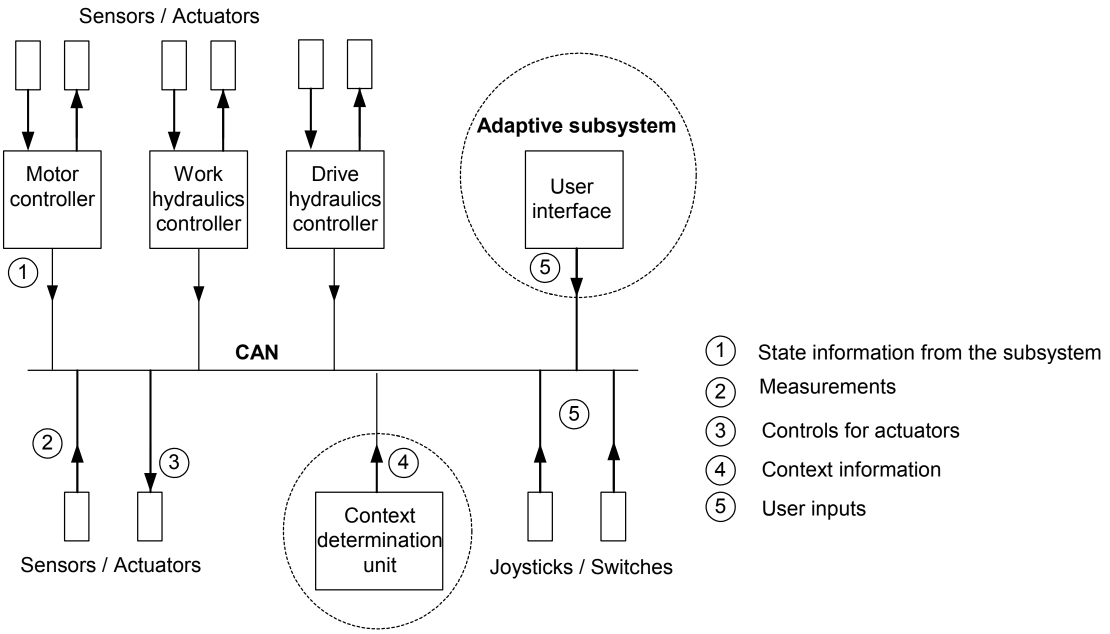


Figure 8. Example of the distributed control system of a modern work machine and a context determination unit added to it.

Distributed control systems are commonly in use in modern work machines. The control system includes subsystems, such as hydraulics controller, drive hydraulics controller, user interface, motor controller, etc. (see Figure 8). These subsystems are connected by a data bus, e.g. a CAN bus. Subsystems are often developed to handle a particular task and their capabilities, e.g. computing power, are selected according to this task. In Figure 8 there is an additional context determination unit. This context determination unit has enough computing power to analyse all the data (e.g. sensor data or status data from the different subsystems) that is transferred through the data bus. The context determination unit tries to derive information about the overall machine functioning from this raw data; it updates the values of the variables that are selected to represent the current context of the machine and periodically sends this data to the other subsystems. The subsystems have their own local rules on how they exploit this context data, e.g. the user interface can adapt itself to the current context. With the context determination unit capable of monitoring the whole system, advanced diagnostic and safety features can also be implemented. In this section a demonstration system based on the concept described above is presented.

4.2.1 Demonstration system

In this research the goal was to study and demonstrate the adaptive and multimodal HMI technologies applied to work machines using a virtual environment and simulation tools. The Envision simulation program was used for the simulation of functions of the work machine and for visualising them to the user.

4.2.2 Case study: Farm tractor used in harvesting

A farm tractor was chosen for the case study because it is a very general-purpose machine used in a wide variety of different tasks in agriculture and forestry work. It is usually used with ancillary equipment mounted on it, e.g. a trailer, a harvester, and agricultural equipment such as a seeding machine or plough. The ancillary devices can be mounted on both the front and the back of the machine, so two different devices can be mounted at the same time. This variety of tasks and ancillary equipment mounted on the machine form a challenge when developing a user interface for a farm tractor. With the adaptive features of the user interface, it is possible to make the farm tractor easier to use with different ancillary equipment and for different tasks.

The task chosen for the demonstration was harvesting with a harvesting boom mounted on the back of the tractor. A virtual environment makes it possible to test control methods that differ from control methods currently in use, so in this demonstration we decided to simulate a highly automated system for harvesting. For this automatic harvesting system a laser pointer is used for pointing at trees for harvesting. The user can use a joystick to control the laser beam and confirm the selection of the tree by pressing a joystick button (Figure 9). Then the automatic harvesting system does the harvesting operations for the selected tree automatically, so the role of the user is to give high-level instructions to the automatic harvesting system and to supervise the functioning of the system. The automation system controls the individual actuators and movements of the harvesting boom in a co-ordinated manner.

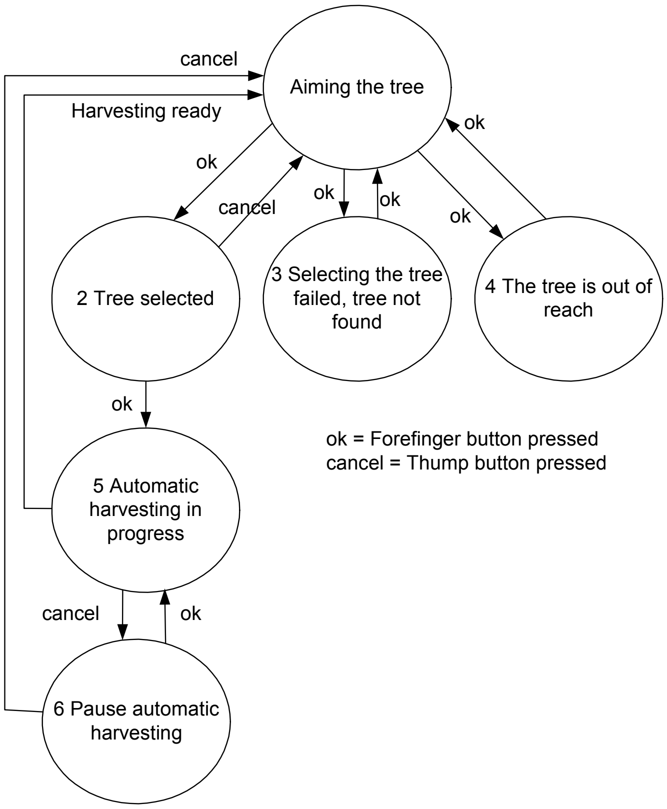

cancel
Harvesting ready
Aiming the tree
ok
cancel
ok
ok
ok
ok
2 Tree selected
3 Selecting the tree failed, tree not found
4 The tree is out of reach
ok
ok = Forefinger button pressed
cancel = Thump button pressed
5 Automatic harvesting in progress
cancel
ok
6 Pause automatic harvesting

Figure 9. A state diagram presenting the operational states of the automatic harvesting system.

4.2.3 Hardware solutions of the demonstration system

The demonstration environment consists of two desktop PCs and an in-vehicle PC with a touch screen (Figure 10). These three PCs are connected to each other via Ethernet. Desktop PC 1 in Figure 10 is for running the Envision simulation program and the virtual model of the machine. An LCD projector was connected to this computer so that the view out of the cabin of the machine could be displayed to the user on the big screen. Desktop PC 2 was used to run applications that manage the context-related data, handle the force feedback joystick and steering wheel, and create sound effects. This PC was also used for program development. The in-vehicle PC was used for running the GUI application.

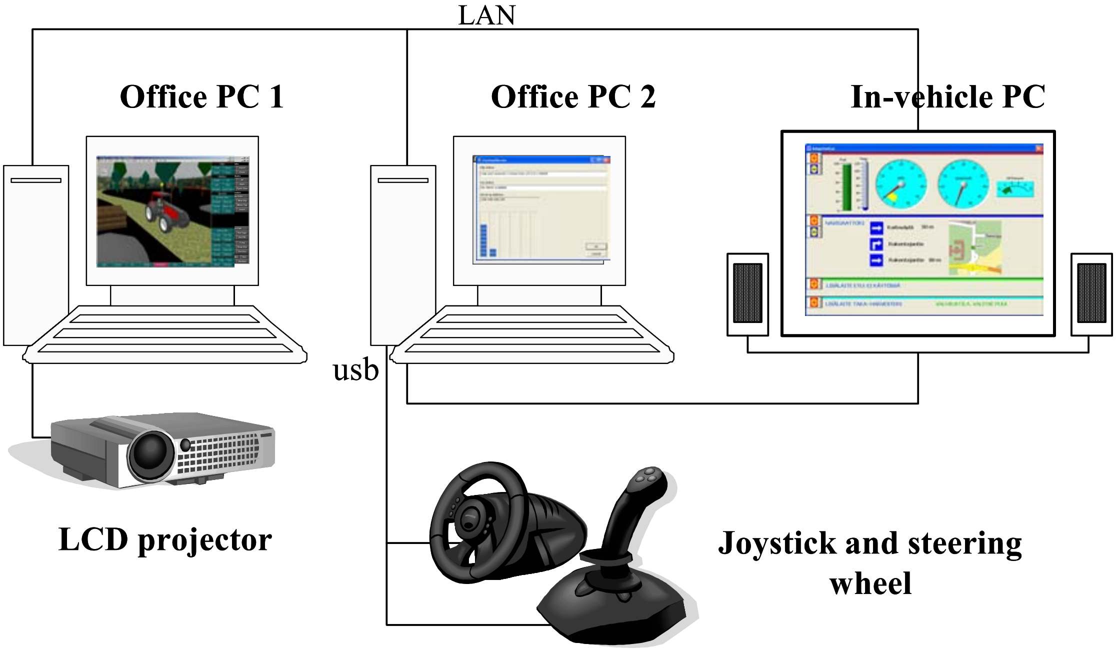


Figure 10. Hardware solution of the demonstration system.

4.2.4 Software solutions of the demonstration system

The software solution of the demonstration system is shown in Figure 11. The software consists of Envision simulator and virtual model (A), Adaptive GUI application (B), FF & Sound UI application (C) and Context server application (D). Communication between these programs is implemented using TCP/IP and UPD/IP sockets and short messages, which have fields for message ID, sender, receiver, data type and data. The Context server application works as an intermediator that forwards messages from sender to receiver. The functionality of these programs and the demonstration system are introduced in the following.

A. Envision simulation program and virtual model of the work machine

The Envision simulation program is used for modelling the farm tractor and the harvesting boom that is mounted on the back of the tractor. The working environment of the tractor is also modelled, as are the trees that are supposed to be harvested. The user can freely drive the tractor in the working environment, in both forward and reverse directions. The driver is also modelled as sitting in the cabin of the tractor. The head of the driver can be rotated using a joystick in order to change the view out of the cabin. The user sees this view on a big screen. The movements of the simulation model, as well as other functionalities of the model, are programmed using the GSL (Graphic Simulation Language) programming language of the Envision program. The GSL program sends data, e.g. state of the boom control system, positions of the actuators and speed of the tractor, to the Context server application. The GSL program receives the positions of the joystick and steering wheel as well as user inputs from the GUI via the Context server application. The movements of the tractor and harvester boom are controlled according to this data.

B. Adaptive user interface

The adaptive GUI application was programmed using Microsoft Visual C++. ActiveX components from the National Instrument Measurement Studio were used for implementing special user interface objects, such as meters and indicators. The Adaptive GUI application receives data, e.g. the speed of the tractor, from the Context server application and shows this data to the user. The Context server also sends a message that contains context information. The Adaptive GUI application adapts itself to the current context according its local rules.

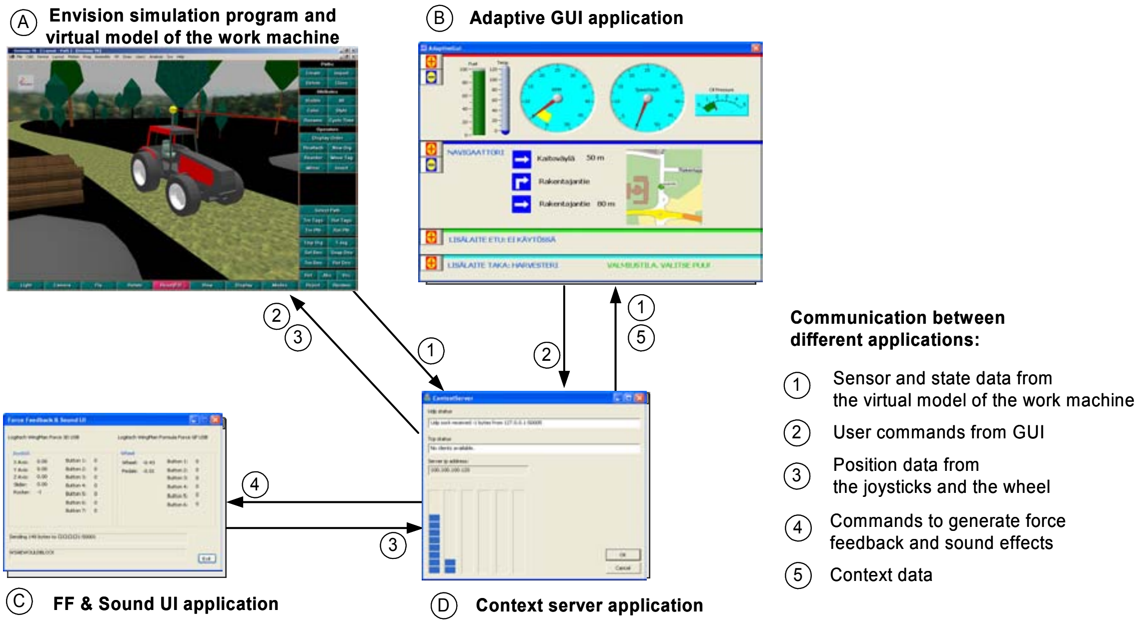

A Envision simulation program and virtual model of the work machine
B Adaptive GUI application
C FF & Sound UI application
D Context server application
1
2
3
4
5
Communication between different applications:
1 Sensor and state data from the virtual model of the work machine
2 User commands from GUI
3 Position data from the joysticks and the wheel
4 Commands to generate force feedback and sound effects
5 Context data

Figure 11. Applications of the demonstration system and the communication between them.

C. FF & Sound UI application

The FF & Sound UI application handles the force feedback joystick and steering wheel. It sends the positions of the joystick and the steering wheel to the Context server application and generates different force feedback effects and sound effects according to the messages it receives from the Context Server application. Recorded spoken messages were used as sound effects in this demonstration. The FF & Sound UI program was programmed using MS Visual C++ and DirectX 9.0c, which is a game programming interface for the Windows operating system.

D. Context server application

The Context server application works as an intermediator that forwards messages from the sender to the receiver according to the sender and receiver fields in the message. So all messages that are sent between different programs in the demonstration system are transferred via the Context Server application. The Context server analyses this data and tries to form knowledge of the overall state of the system. It updates the values of the variables that are selected to represent the context of the system and sends this data in a message to the *Adaptive GUI application* (B). It also detects situations when a force feedback or a certain sound effect should be generated and sends a message to the *FF & Sound UI application* to generate the desired effect.

4.2.5 Context determination method

Context determination in the work machine applications is based on processing raw data from the machine's sensors and subsystems into meaningful variables that represent the current context of the machine. Context information in this case is a selection of variables that represent the current context of the machine in a usable form. Adaptive subsystems can use this data and adapt their functionality according to it. The selection of member variables that form a context of the machine is very application-specific – for example, it depends on the sensors that are available in the machine. The values of the context member variables can be directly obtained from the sensors or derived from raw sensor data. In this demonstration the following variables were selected to represent the context of the work machine:

x1 use rate of the joystick
x2 use rate of the steering wheel and the foot pedals
x3 use rate of the GUI
x4 speed of the machine
x5 use rate of the ancillary equipment mounted on the front of the machine
x6 use rate of the ancillary equipment mounted on the back of the machine.

The member variables *x1*, *x2* and *x3* give information about the activity of the user and about the task he/she is currently conducting. The member variables *x1* and *x2* are derived from the positions of the joystick and the steering wheel, and the changes in positions. Variable *x3* is determined by counting user inputs via the GUI. The variables *x4*, *x5* and *x6* give information about the machine's current operational status and task.

All the member variables are scaled to values from 0 to 100. For example, *x1*=100 means that the user is using the joystick very actively, or *x6*=100 means that the ancillary equipment on the back of the machine is currently in active use. This context information that contains the values of these six variables is sent to the adaptive subsystems, which have their own local adaptation rules. In this case the context information is sent to the adaptive GUI application.

4.2.6 Adaptive features of the GUI application

The GUI design used was based on categorising the functions of the GUI into four different groups based on the functions of the machine. The layout of the GUI was developed based on this grouping of functions. The four categories are:

1. Basic instruments needed for driving the tractor
2. Additional functions related to driving, in this case navigation system
3. Functions related to the ancillary equipment mounted on the front of the tractor
4. Functions related to the ancillary equipment mounted on the back of the tractor, in this case a harvesting boom.

The layout of the GUI consists of four panels, one panel for each group of functions. There are three alternative choices for the size of each panel – minimum, medium or maximum size. These four panels can be fitted to the screen using two different layouts:

- Layout A: One maximum size panel and three minimum sized panels
- Layout B: Two medium sized panels and two minimum sized panels.

Because all the four panels are shown at the same time, the user always sees the most important functions of the machine and can easily access the different functions of the machine. The panels can be made smaller or bigger, so the amount of data shown on the screen can be scaled (Figure 12). Adaptive features were implemented in the GUI so that it automatically selects the right size for the panels according to the context data. The problem of selecting the correct sizes for the four panels was solved as follows:

1. Priorities *p1…4* are calculated for each of the panels:

$$p_a = \sum_{n=1}^{6} w_{an} \cdot x_n$$

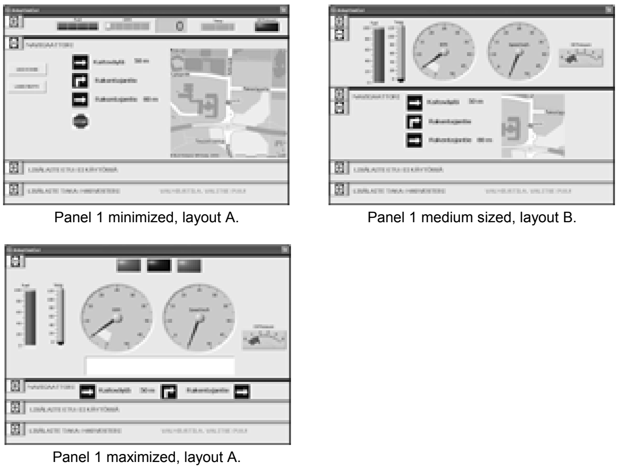
where: *a* is the number of panels 1…4
wa1…6 are weighting factors
x1…6 are member variables of the context.

2. The panels are arranged according to the priorities *p1…p4*.
3. The layout of the GUI is selected by comparing the two panels getting the highest priorities. If the priorities are near to each other, layout B is selected, otherwise layout A is selected.
4. If a change in the panel sizes occurs, the GUI is updated.

The functioning of this method is fine-tuned by changing the weighting factors *w1…6*. Also, more complex expressions can be used for calculating priorities *p1…4*. Other methods for solving this problem are to use a set of rules or fuzzy logic (Mäntyjärvi 2003). The straightforward method presented here was seen as adequate for demonstration purposes. The user was also given a chance to change the sizes of the panels manually. After a set time period (10 sec) the GUI application returned to the automatic adaptation mode.

Another adaptive feature of the GUI was that the information shown in the panels was changed according to the current context. Here are some examples of how the adaptive GUI works in different situations:

- When the user is mainly using the steering wheel and the pedals, the system maximises the size of panel 1, which shows the meters and indicators related to driving.
- When the user is mainly using the harvester, panel 4 is maximised, and the graphics of this panel are changed according to the status of the automatic harvester control system.
- If the user drives the tractor and simultaneously uses some ancillary equipment, panel 1 and panels 3 or 4 are shown medium sized.
- When the speed of the tractor increases the navigational panel is diminished and when the speed is low the panel is enlarged.



Panel 1 minimized, layout A.

Panel 1 medium sized, layout B.

Panel 1 maximized, layout A.

Figure 12. Adaptive GUI and different sized panels.

4.2.7 Force feedback effects

The force feedback joystick and the steering wheel that were used in the demonstration system were low-cost devices typically used in computer gaming. Force feedback effects were implemented using a DirectX programming interface. Force feedback was used to warn or inform the user about exceptional situations. For example, when the user was harvesting trees by pointing at the trees with the laser pointer, force feedback effects were used if the pointed tree was out of reach or the pointer didn't hit any trees, or if some of the actuators exceeded their operating limits. The force feedback was seen as an effective way of getting the user's attention, especially if noise prevents hearing auditory signals. A drawback was that in work machine applications the user's hands are not always on the joystick and if there is a lot of vibration, the force feedback effect can be hard to notice. Also, recognition of the different force feedback effects proved to be harder than expected. Because of these facts, supplementary visual feedback was also used, – e.g. if a limit of an actuator was exceeded, the link exceeding the limit was changed from green to red in the graphics presenting the harvesting boom. More accurate sensing of different force feedback effects would have been achieved by using better quality force feedback devices.

4.2.8 Sound effects

The sound effects used in the demonstration were recorded speech messages. There were thirteen different messages, e.g. "Tree too far", "Tree not found" or ”Harvesting ready". DirectX was used when generating the sounds because it enables playback of multiple sounds simultaneously and with a very short delay. The sound effects were used to inform the user about abnormalities and state changes in the system. In this demonstration the user could observe the operation of the automatic harvesting system by listening to the speech messages. Speech messages were seen as a strong method for capturing the user's attention. In the demonstration the same information was also shown on the screen. This ensures that the user is aware of the status of the system even if he/she doesn't hear the sound message or he/she continues working after a break. Too intensive use of sound messages can irritate the user, so sound messages should be used carefully. In work machine applications the volume of the sound messages should be adjusted automatically according to the noise level while working.

4.3 Results

The object of this research was to study context awareness and adaptive user interfaces in work machine applications. A demonstration system was developed for that purpose.

Context determination is based on collecting data from the user, the machine and the environment. The values of the variables that are selected to represent the context are derived from this raw data. The selection of the member variables of the context is challenging and often a very application-specific task, and depends on the available sensor data.

In the demonstration system the user was able to use the virtual model of the work machine, in this case a farm tractor equipped with a harvesting boom, using joysticks, a steering wheel, pedals and a graphical user interface. The variables selected to represent the context of the system were user activity and different operational states of the machine. The determined context was used for adapting the graphical user interface to the current situation by changing the layout of the GUI and the amount of data presented in the GUI.

The user interface of the demonstration system also consisted of sound effects and force feedback effects. These techniques were found useful as supplementary methods for the graphical user interface. Some drawbacks were also noticed, e.g. different force feedback effects were not easily distinguished from each other and sound effects could easily irritate the user. These facts limit the usefulness of these methods in machine control applications.

5. New Product Development

5.1 Introduction

Companies are diverse with different commercial activities, industrial interests, and ways and means to produce innovation, products and services. The roles of production and use produce a life cycle, e.g. the working machines or their cabins are in a state of continuous change. This change means the focus shifting to core competence, networking and human technology interaction. Moreover, design outcomes are no longer bare manufacturing but also capacities to produce, work out, serve, make decisions or manage things. Common benefits that companies would like to achieve are competitive advantage, ability to be responsive to customers' needs, increase in profitability, increase in total quality, reduced cycle time in bringing a new product to market, company-wide feeling of accomplishment, or continuously improving the company.

Companies are also interested in the product's sustainability throughout its entire life cycle. Therefore, data and information gained from the products in operation is relevant for improving operational dependability, systemic usability and overall cost-effectiveness. In general, the products are composed of more complex systems with computer-mediated interfaces partially hiding the actual job(s) during operation. Indeed, user work has become more demanding with concurrent multitasking and multipurpose interfaces and instrumentations. Designing such products is complicated, and necessitates exactness and carefulness but also the ability to balance resources between numerous tasks and functions, and profoundness in relation to relevant demands.

However, during recent decades the designers' tools have increased in number and become powerful in many ways, which, in principle, allows more time to fulfil the demands of design and development assignments. Similarly, the possibilities of manufacture in companies have widened. Despite the novelties the technologies afford, the product development as a process or several processes in a real company has become much more complex and cross-linked to functional phases and sub-phases of the processes and sub-processes. Sometimes this process is difficult to manage; sometimes it makes the company's activities firm and bureaucratic.

5.1.1 Challenges

Technological product properties, even in different situations during the life cycle, can be simulated by applying various design models – first, to ensure system level functionality and feasibility; second, to provide sufficient data for the design of further details; and finally, to test the product. Here, virtual prototyping can be used to evaluate and verify the design output. Digital human models can be used to design and evaluate the ergonomics and usability of the product. Various virtual reality-type approaches, such as virtual environment appliances, can be used at the beginning of the process to obtain and collect user information for later design phases – e.g. for concept design, system architecture design, and details design. Virtual environments may also be utilised to verify the product design and even to partially validate some aspects of the product without actually having a real product.

However, the procedures on how to fully exploit these new tools in product development need further development, or even a totally new way of thinking, in order to be applied to the product process and related processes.

Manufacturing technology can be similarly designed and tested in advance to gain fast ramp-up and minimise costs. An estimation of overall performance, etc., in different situations the product may encounter is possible, in principle, using modern technological enablers. At present, there are no feasible methods and practical procedures on how such virtual reality-based systemic design approaches can be implemented to provide the product process with advanced technology enablers for everyday product design and development in a controlled way.

5.1.2 Objective and limits

The objective for new product development in this study was to clarify the issues from the product development process point of view, and the relationships to other relevant processes, in order to introduce the developed virtual reality-based system and the developed methods for cabin design.

However no practical implementation of the developed system was tested at company sites. Thus the results are based on tests carried out using laboratory use cases in co-operation with the researchers and designers in the workshops.

5.2 Generic product development

A process is defined as a series of actions or operations conducing to an end, or a sequence of steps that transforms a set of inputs into a set of outputs. In contrast to a project, a process is not unique but it continuously reproduces itself. Additionally, the process does not include the resources of the activities. However, it has an appointed owner, management and steering board, and they organise, initiate, develop and control the activities and functions of the process.

Here the special interest was in the product development, applying simulations and virtual reality-based approaches to do the cabin interfaces design; by using the applications of computational models to the study and prediction of physical events or the behaviour of working machine-types' engineered systems. Simulation and virtual reality represent interesting extensions to design as they are based on mathematical models. Such models attempt to characterise the physical predictions or consequences of scientific theories. Simulation also provides a powerful alternative to the techniques of experiment and observation when phenomena are not observable or when measurements are impractical or too expensive (Tinsley et al. 2006).

5.2.1 Innovative products

The new products are more and more based on computer-mediated interfaces, which has increased the importance of the innovation process as one of the key processes in new product development, although it is not always mentioned as an innovation process. Green et al. [1995] divide the characteristics of innovation into categories. This indicates that innovation can be thought of as being comprised of at least four distinct, independent items: (1) Technological uncertainty; will the technology itself work as planned?, (2) Technological inexperience; does the firm know enough about the innovation?, (3) Business inexperience; is the firm familiar with the market and industry?, and (4) Technology cost; what is the overall cost of the project?

5.2.2 Issues of new product development

New product development and the success factors have been studied widely. Apilo (2004) suggests that combining the strength of product development projects with other development projects could bring benefits at the company level through both efficient project management and systematic change management.

Companies usually have several development projects running contemporaneously. They also have a lack of resources to implement change as a consequence of the numerous projects. Thus projects should be put more carefully against the company's strategy. The definition phase of strategy implementation projects starts during the strategy process. Because of the complexity of strategy implementation, there is a relevant need for effective project management methods.

However, development projects are not alike. According to Apilo's (2004) studies, there are plenty of differences between project durations; product development projects had been carried out in one to five years and product improvement from two months to one year; most process development projects had been carried out in one year, but information technology projects in two to three years. The difficulties in running projects are encountered because of unclear project scope, lack of resources due to daily business work, or lack of top management sponsorship (Ulrich & Eppinger 2002, Apilo & Taskinen 2006).

According to Juran (2001) success in achieving business goals and objectives heavily depends on large, complex, cross-functional business processes such as product planning and product development, invoicing and customer care, and purchase, procurement of materials, production and product distribution. In a constantly changing environment without management attention over time, the processes become obsolete, over-extended, redundant, excessively costly and ill-defined, and have loose adaptability to the demands of the environment. Nishiguchi (1996) emphasises the importance of (1) Market learning and technology learning in co-operation, (2) Decision integration between functions, (3) Integration of technology and business strategies, and strategic consistency, (4) Differentiated self-structuring organisational contexts for projects, and (5) Cross-functional linkages, team empowerment and teams' relative autonomy.

5.2.3 Lifecycle processes related to product development

The ISO/IEC 15288 Standard on System Lifecycle Processes deals with system lifecycle processes, hardware, software and human interfaces. The framework of this standard encompasses the lifecycles of man-made systems. It starts from the conception of the ideas and continues through the different stages to the retirement of the system. It provides the processes for acquiring and supplying the system products and services that are configured from one or more of the following types of system components: hardware, software and human interfaces.

Moreover, this standard also provides a framework for the assessment and improvement of the project lifecycle. This forms a comprehensive set to construct lifecycle models appropriate to the product, service types and markets in which they trade. It may also be used, via contract or agreement within the supplier/acquirer relationship, to select, agree on, and perform the processes and activities called out in the standard. Generally, products or systems configured from different types of system components have a lifecycle.

The lifecycle may vary a lot in accordance with the purpose, design, production and use. However, there is a general set of characteristic phases or stages that make up the complete lifecycle of any system. These represent the major periods that are frequently encountered and associated with it and are related to the state of it (Figure 13).

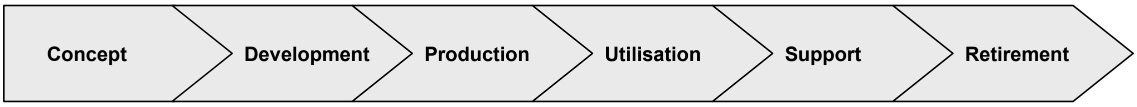


Figure 13. Diagram of system lifecycle processes adapted from ISO/IEC 15288.

Here, the stakeholders' needs are identified during the concept stage, the concepts are explored and viable solutions are proposed. The development stage engineers a product that is a producible item; the stage encompasses refinement of requirements, description of the solution and construction, verification and validation of the product. The production stage manufactures the item and confirms design verification. The utilisation stage considers operation-related activities, and the support stage takes care of maintenance and support issues and activities, and both give validation related information and feedback. Finally, the retirement stage considers retiring, disposing and archiving. The above stages also provide organisations with a framework within which management may have high-level visibility and control of the product.

5.2.4 Relationship between utilisation and maintenance in product development

There are differences between products. Software product lifecycles tend to have different durations compared to physical products: the initial, concept stage seems to be more demanding and the production is actually a minor stage but continues during the utilisation stage concurrently with the operational evolution of the product (Figure 14). Engineers have traditionally considered any work after initial delivery as maintenance. Even this work has been divided into various tasks, including making changes to functionality (perfective), changing the environment (adaptive), correcting errors (corrective), and making improvements to avoid future problems (preventive).

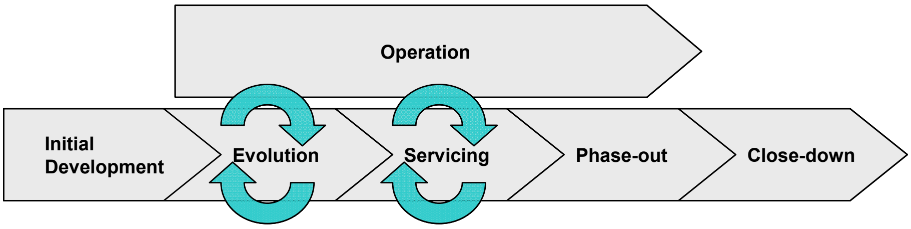


Figure 14. Software products have different, more use-oriented lifecycle processes (adapted from Rajlich & Bennett 2000).

Products have different stages. For example, the stages of a paper machine differ considerably from those of a joystick, or from those of software. There also seems to be a change from bare technology-driven products to more user-driven products, and towards use-driven products. The design and development activities are also needed more often during the product's lifecycle but are divergently initiated from the utilisation and support stages of an existing product; often called re-designing.

Maintenance has been considered basically uniform over time, but some believe this approach no longer suffices. A new view of the software lifecycle in which maintenance is actually a series of distinct stages, each with different activities, tools, and business consequences, has been described. However, industry still considers post-delivery work simply software maintenance, but authors claim that the process actually falls into stages.

This is especially the case in the use-driven products' development, where there is also certain diversity in understanding these stages and their transitions.

During recent years, the increase in use-driven products has extended the duration of design and development to the commissioning stage and even to the later stages. This is because it has been impossible to finish all the design and development during the common product process. For example, although the product is already in use, the complex interface/interaction systemic details need immediate redesigning according to plan, but the actual facts for that are only known then. Because of that, the product process is extended, both to the start and to the end. The use of modern knowledge management and novel design tools have made knowledge transfer easier and transparent, and thus made it possible to extend the concurrent coverage of design iteration. On the other hand, the innovation process has been introduced in companies to manage this kind of inter-process activity in initiatives.

5.2.5 Product lifecycle representations

The systematic approaches to the design process are structured in two dimensions (Roozenburg & Cross 1991). The vertical dimension of models divides the evolution of a product into life phases. The horizontal dimension is based on a basic cycle of problem solving that takes place in every phase of the vertical structure. The basic cycle of problem solving represents a generalised way of describing the horizontal dimension.

Thus Pahl & Beitz (1995) include similar activities as in the ISO/IEC 15288 standard in their product lifecycle representation. The relationship between product lifecycle management, project plan, and product goals and requirements, or specifications, is shown in Figure 15.

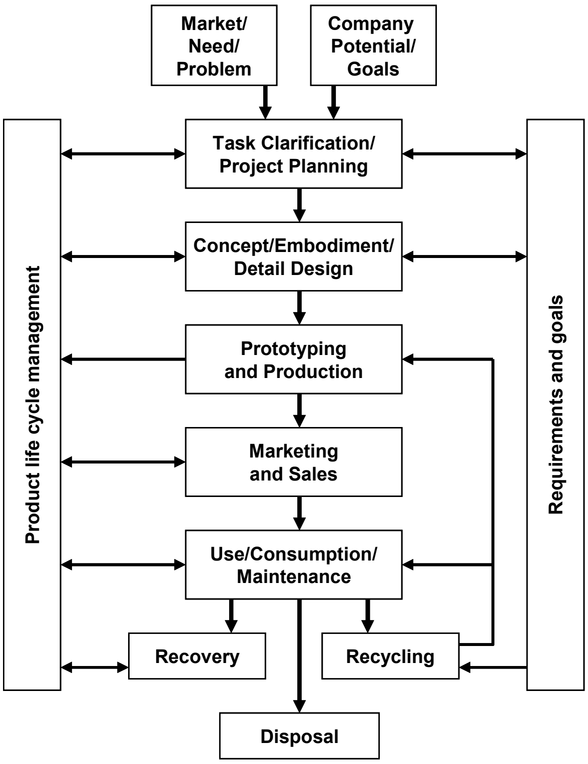

Market/ Need/ Problem
Company Potential/ Goals
Task Clarification/ Project Planning
Concept/Embodiment/ Detail Design
Prototyping and Production
Marketing and Sales
Use/Consumption/ Maintenance
Recovery
Recycling
Disposal
Product life cycle management
Requirements and goals

Figure 15. Product lifecycle (adapted from Pahl & Beitz 1995).

Another way of characterising the horizontal dimension is to describe the various methods and techniques a designer uses in each stage of the design process.

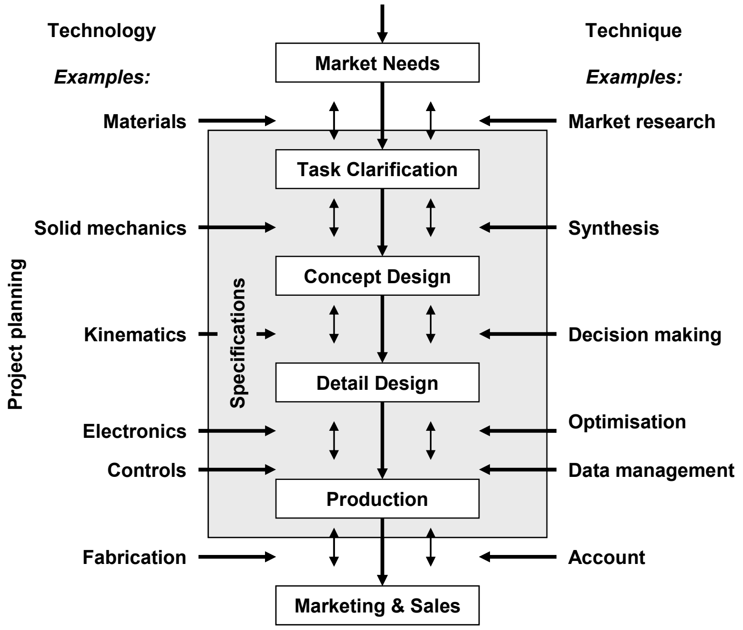


Figure 16. Total design activity approach (adapted from Pugh 1991).

Pugh's (1991) total design activity approach (Figure 16) is a more extended representation having the product development process with central design core activities. It has similarities to the activities in many previous process representations. The product specifications are shown surrounding the design core. Discipline-independent techniques are shown as inputs on the right-hand side of the design core. Here, as examples, are market research, synthesis, decision making, optimization, data management and accounting, shown as joining near to concepts. Technology from various disciplines is shown as inputs on the left-hand side of the design core, with its own type of examples.

5.2.6 Generic development processes

The generic development process introduced by Teakle & Gilmore (1999) is broken down into five sequential phases (Figure 17), each having iterative loops.

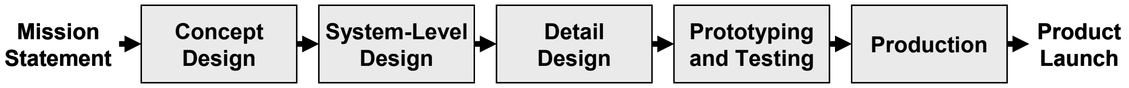


Figure 17. Generic product development process (adapted from Teakle & Gilmore 1999).

Here the concept development stage identifies the market needs, does the task clarification, project planning, concept generation and selection, and finally combines the information for the following stages. The system level design stage establishes the product architecture and defines ways in which the various functions of the product are implemented in physical subsystems and also how they interact with each other. The detail design stage works out complete specification of parts and takes account of the tooling required for making the parts. Process plans for fabrication and assembly are also established. Testing and refinement involve the construction and testing of pre-production prototypes. Production ramp-up defines how the parts are manufactured and assembled using the intended production system, and the production system is fine-tuned. Finally, the refined product is launched and made available for distribution.

However, these generic product development process phases are not always the case because of the great variety and diversity of different product design cases. Considering the new product development, the first functioning versions of the product concepts are usually developed for testing purposes from scratch to satisfy the initial requirements that are set up in the planning and concept development phases.

Ulrich and Eppinger (2002) define the product development process as *"the sequence of steps or activities which an enterprise employs to conceive, design, and to commercialise a product"* (Figure 18). Thus the product development process is basically composed of the first two lifecycle stages: the concept and the development stages.

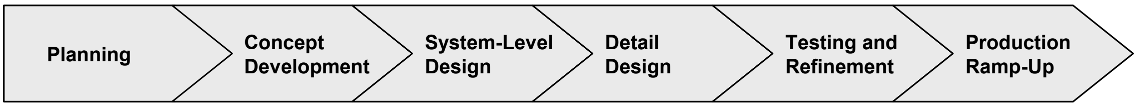
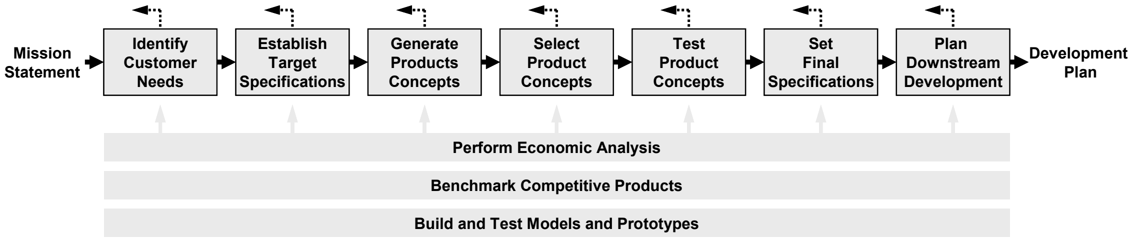


Figure 18. The generic product development process (adapted from Ulrich & Eppinger 2002).

Here, the planning phase corresponds to the above-mentioned mission statement and precedes the project approval and launch of the actual product development process. The company's strategy is responsible for it. It includes an assessment of the technology developments and market objectives. The output will be the mission statement: target market, business goals, key assumptions and constraints. The concept development phase producesthe concept, description, function and features of the product, and an economic justification for the project. The system level design phase includes a geometric layout of the product, a functional specification for the subsystems and a preliminary process flow diagram for the final assembly process. The detail design phase produces the control documentation for the product, a description of the geometry of each part, production tooling, specifications of the purchased parts, and process plans for the fabrication and assembly of the product. The testing and refinement phase involves the construction and evaluation of multiple pre-production versions of the product; early alpha, alpha, beta and later beta prototypes. Prototyping presents a thorough discussion of the nature and use of prototypes. The production ramp-up phase makes products using the intended production system and carefully evaluates any remaining flaws. The transition to launched production is gradual, and the product becomes available to widespread distribution. Each of the above phases is composed of sub-phases, as shown in Figure 19.



Mission Statement
Identify Customer Needs
Establish Target Specifications
Generate Products Concepts
Select Product Concepts
Test Product Concepts
Set Final Specifications
Plan Downstream Development
Development Plan
Perform Economic Analysis
Benchmark Competitive Products
Build and Test Models and Prototypes

Figure 19. Sub-phases of the concept development (adapted from Ulrich & Eppinger 2002).

5.2.7 Transfer of new products to the market

The development of new products also has different features compared to the re-design type of product development that happens with no or only minor features of the concept stage. Moreover, it seems that companies are dependent on their ability to transfer new products within the company into the market successfully. Brethauer (2002) emphasises that only 10 per cent of all new products are successful in the market. Thus this 10 per cent of releases are responsible for almost half of companies' annual revenue, and 90 per cent of new products never produce revenue. Companies waste valuable time and money developing good products that never see the light of day because of the lack of an effective internal transfer process.

Successful transfer of new products through the organisation is a common problem throughout industry (Brethauer 2002). New product transfers internally are accomplished serially in many companies; the activities are completed in one department at a time and then passed on to the next department with little or no communication, which does not work well.

Brethauer (2002) suggests developing an internal product transfer process that (1) integrates all departments into the process at the same time, yielding a gap-free transfer, (2) builds on solid tools and techniques that streamline execution and produce effective results, and (3) is developed and implemented such that the whole company embraces the transfer process. The company thus has both strategic goals and tactical programs to accomplish the goals and will learn how to accelerate their new product launches by using the new product transfer process. The stage-gate methodology (Figure 20) forms one of the selectable tools to reduce the cycle time and improve new product success rates.

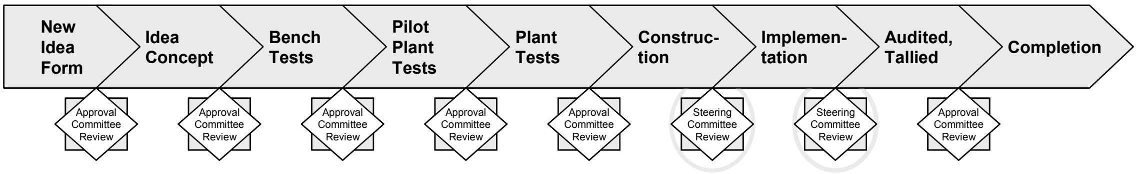

New Idea Form
Idea Concept
Bench Tests
Pilot Plant Tests
Plant Tests
Construc-tion
Implemen-tation
Audited, Tallied
Completion
Approval Committee Review
Approval Committee Review
Approval Committee Review
Approval Committee Review
Approval Committee Review
Steering Committee Review
Steering Committee Review
Approval Committee Review

Figure 20. State-gate methodology-based product transfer process diagram (adapted from Brethauer 2002).

Brethauer's (2002) keys to successful product transfer are: Clear objectives, Cross-functional team approach, Upper management support, Customer-supplier involvement, Robust design and product, Structured process methodology, Economic justification. The first four are communicative tools, the second two are robust design and product tools, and the last one is a financial tool.

Communication will get better by establishing core teams, or cross-functional teams, gathering workers from all major divisions or departments within the organisation. Usually, these members are leading project teams in their specific area and are the key communicators with the people in their functional areas.

The teams must have clear goals and objectives as a first task, which become the criteria by which the team developing the product will be judged. The objectives should be clear and be specific, measurable, achievable, realistic, and time-specific in order to be useful and assessable. An example: *"The new surveillance camera must perform 24 hours per day with only 2 per cent downtime. This new enhancement must be released by the fourth quarter this year"*.

5.2.8 Simulation-based new product development process

Utilisation of computer-based methods such as modelling, simulation, virtual prototyping, virtual environments, virtual reality, etc., has increased rapidly during the past few years. Their advantages are manifold, not only do they reduce physical prototypes, mock-ups and number of tests but they also increase quality, ergonomics, and usability.

Sääski (2006) has studied the several product development methodologies starting from Hansen's Konstruktionsystematik (Hansen 1965) and following the development steps up to this date (Andreasen & Hein 1987, Pahl & Beitz 1977, Pahl & Beitz 2003, Hubka 1987, Roth 1981, Tjalve 1979, VDI 2221 1987, Pugh 1991, Pugh 1996, Suh 1990, Ulrich & Eppinger 1995, Ulrich & Eppinger 2002, Cagan et al. 2002). The actual change in the content of the stages, or phases, since the 1980s is minor and the differences in the theoretical representation are also minor, such as different emphasis on single stages.

However, the development of the tools and applications a designer may use during her or his design tasks is significant. The significance of the compatible product development process will increase even more when the product development is enlarged to contain more and more of the lifecycle stages and their phases – e.g. manufacturing, utilisation, maintenance and support, disposal and recycling.

Sääski (2006) introduced a new generic product development process based on interviewing the globally operating Finnish design project leaders. The composed and developed new product development process was also verified (Figure 21). Each of the six phases of the process representation also includes more precisely detailed process representation.

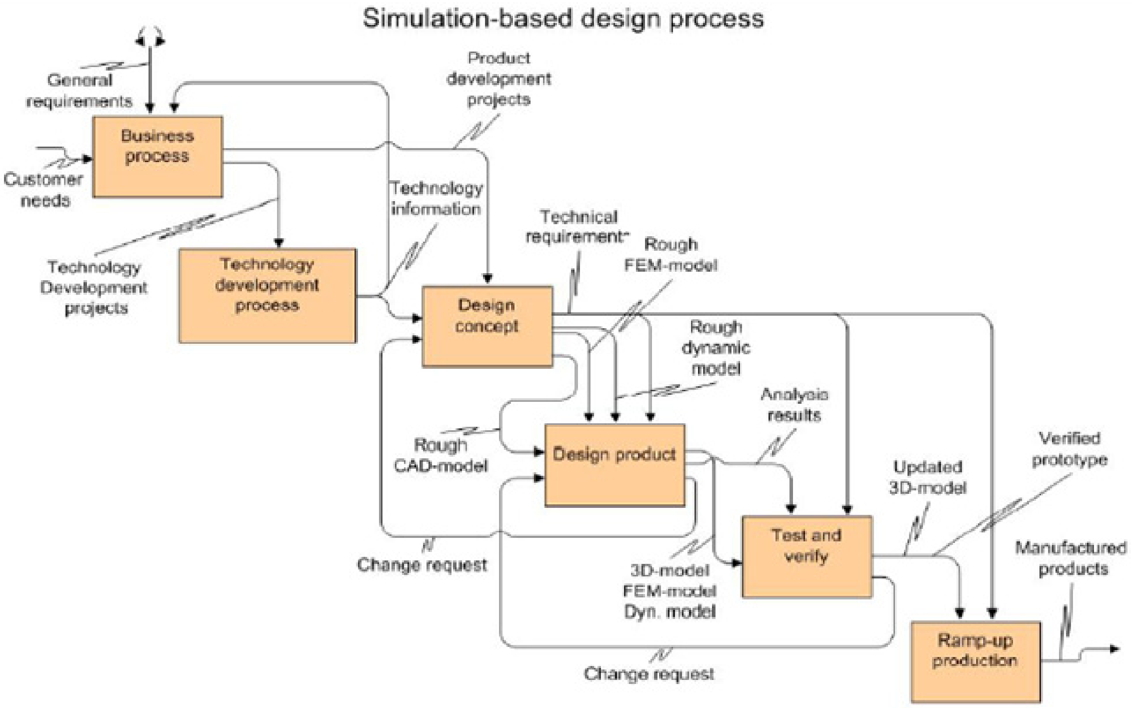


Figure 21. The simulation-based new product development process (Sääski 2006).

The operators of working machines can be used to obtain information on user requirements, product design demands, interactions and relevant relationships with the active systemic environment. The design process needs to be able to cover the whole system (cf. EN ISO 6385). Product developers have to comprehend the whole field of human work and activity. This means that 3D models and simulation are needed to identify and recognise human physical and cognitive needs in a working environment.

Simulation also provides possibilities for both collaborative and participatory design approaches. Human-centred design methods (cf. EN ISO 13407) need to be individually developed and adapted to the company's product development process at the organisational level so that they better support today's concurrent and distributed design processes. Specifying the technical requirements for the product or system is a major task. Thus in human-centred design it is essential to extend these technical requirements to include human physical and cognitive demands too.

The product development will predictably produce successful design solutions. The process therefore involves using existing knowledge (standards, guidelines examples of other systems, etc.) to develop a proposed design solution. Thus it makes the design solution more concrete by using simulations, physical prototypes, mock-ups, etc., showing the product prototypes, observing the performance of specified tasks, using this feedback to improve the design and iterating this process until all the design objectives are met. Human-machine interaction simulation uses 3D models, the biomechanical model and the environment model to simulate human operation.

Sääski (2006) introduced different process phase representations into product development, e.g. the human-machine interaction process phase (Figure 22).

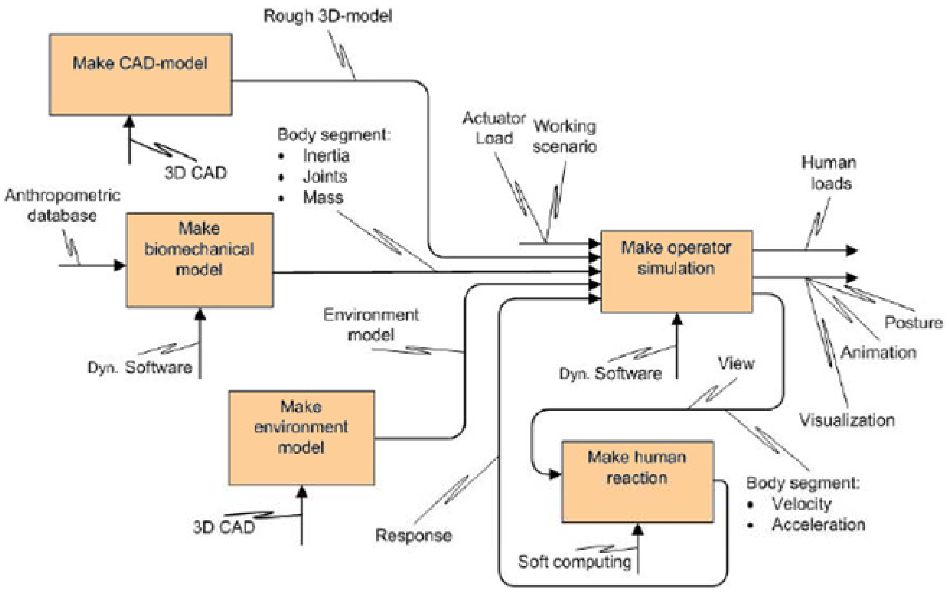


Figure 22. Human-machine interaction process phase representation (Sääski 2006).

5.3 Results

A concept model for making the relevant linkage to the company product development process was investigated. The actual product development processes comprise two or even three rather different types of functional process entities: the planning phase, the concept development phase and, later, the product development phases.

The importance of this potential division of the product development process is that it may create diversity and incompatibility for the selection, implementation and use of tools, applications, methods, etc., and thus also inefficiency, prolongation, reduction in quality and extra costs. All the disadvantages may slow the product transfer into the market, which also has extra economic drawbacks. Therefore, care must be taken when executing the development and implementation of the product development process as the product transfer-related issues of the processes are of great importance.

Communication, knowledge transfer, cross-functional teams, etc., have to cooperate fluently both internally and externally. Knowledge sharing is a significant factor in facilitating the communication and boosting the work intensity and productivity. Common interests and compatible procedures, tools, applications, data, etc., also foster integrity and consistency of company activities.

However, profitability is dependent on several things besides functionality. There are variants in the generic development process during the initiation of the process: generic (market pull), technology push, platform push, process-intensive or customised. This variation is in accordance with a company’s unique context. The concept development phase of the development process demands more coordination among functions than any other phase. It is necessary to have enough valid information in every situation in order to make the correct strategic decisions on products, manufacturing, markets, delivery, etc., because everything else in the operation is founded on it.

The realistic living product development process is much more complex in contrast to what can be represented with a simplified generic product development process model. In reality, the product development process is some kind of complex compilation of innovation process, product development process, production process, support process and commission-delivery process. Therefore, the descriptions above must always be carefully thought through before any implementation, reasoning the company’s strategic base of abilities and competencies, and the relevant choices available to develop the company’s processes and activities. As a process owner, you have the right to develop and make changes to details. However, you are most probably strongly bound to cross-linked duties that may also be shaped by the ‘individual’ sub-processes owned and managed by other people; those sub-processes may be linked to other upper-level processes too. Therefore, in practice, the hierarchy and cross-links of the processes should not be too extensive to avoid bureaucratic features that may hinder the success of the management.

Success is based on the balance of the whole, because the processes and activities are cross-linked. Figure 23 presents some design and product characteristics and areas of focus along the lifecycle of a product. A feasibility trend is also given, which characterises the feasibility of the developed virtual reality system. The feasibility decreases along the lifecycle as a consequence of the increasing complexity of the design functionalities, details and model features, and the increasing amount of cumulative data.

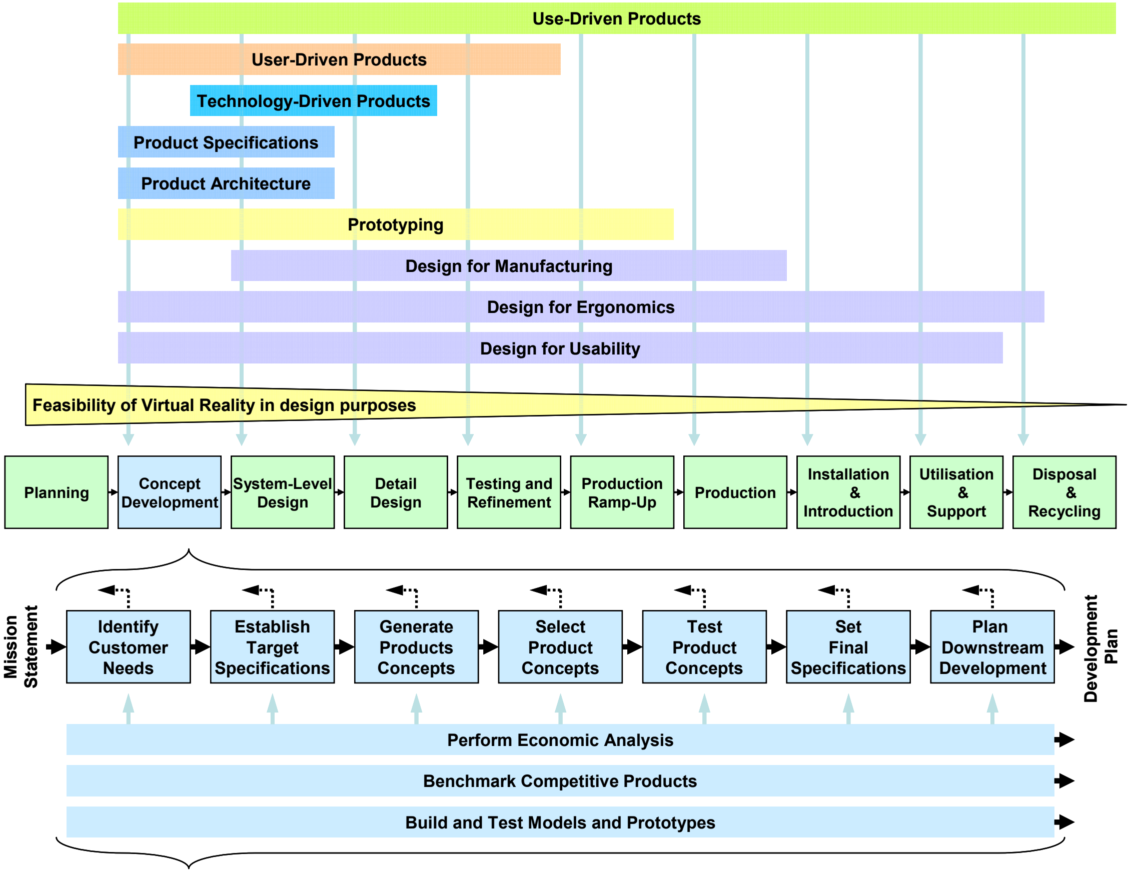


Figure 23. Product development process and phases are cross-linked to other processes and the product's lifecycle in a complex way. Thus it is not possible to easily represent a generic product development process model (features adopted from Ulrich & Eppinger 2002).

6. Case Studies

The purpose of the case studies was to evaluate the future development methods described in Section 2.5. The case studies were evaluated in industrial workshops by designers from vehicular working machine companies, human factors experts and researchers.

6.1 Case Studies and Industrial Workshops

Two real products were used in the case studies: a fork lift (Figure 24) and a tractor (Figure 25). Even though the case study products are existing vehicular working machines, the working processes are theoretical. There are five different working processes in the case studies. All the selected processes can be tested with both products in VEs. These working processes are as follows:

1. Cutting trees with a harvester
2. Picking up objects in a warehouse
3. Spadework
4. Driving with a trailer
5. Tillage.

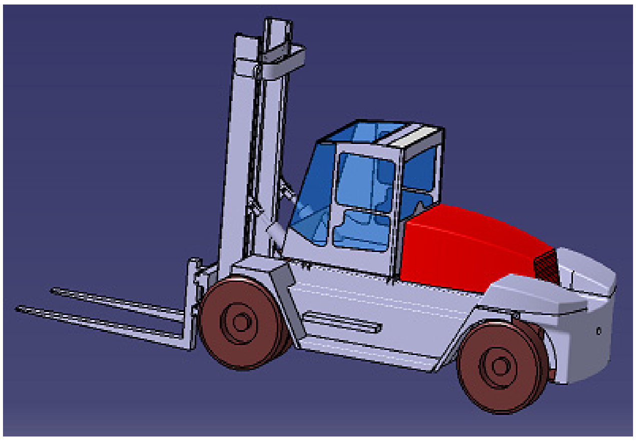


Figure 24. One of the case study targets: a fork lift.

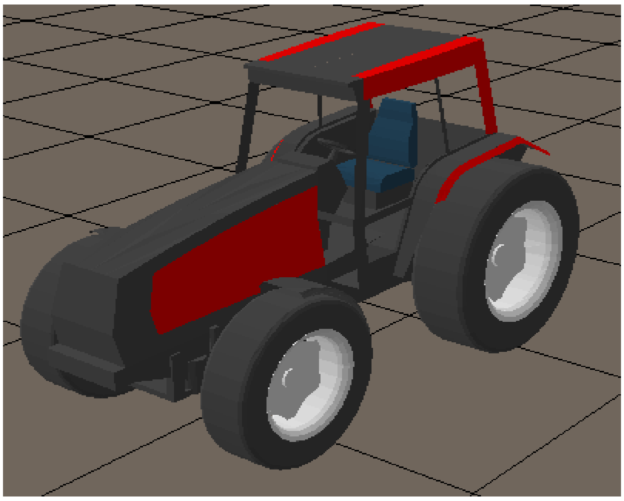


Figure 25. One of the case study targets: a tractor.

In this study the four-step user-centred design method and participatory design method were utilised and tested by designers from vehicular working machine companies, human factors experts and researchers in industrial workshops. The designers used the four-step design method for user interface design in VEs. The main goals of the industrial workshops were to test the method, get feedback from the designers and commit the designers to the use of the new four-step user-centred design method.

Five industrial workshops took place over eighteen months. The first workshop was hosted at VTT's VE laboratory (described in Section 2.5). The first version of the four-step user-centred method was ready and was exploited using a one-wall passive stereo VE system with a flexible control system (described in Section 2.5.1). The second workshop also took place in VTT's VR laboratory. The main focus of this workshop was on utilising the first and second steps of the four-step method. The method was exploited and tested in tractor cabin design and the work process was tillage. The main focus of the VE system was on the interactive control system and digital human model. The third workshop involved the use of a CAVE-like VE at the VR laboratory at Seinäjoki University of Applied Sciences, the School of ICT. The CAVE was a five-wall active stereo system. The main focus of the workshop was on testing CAD-like commercial VE software and testing the second and third steps of the four-step method. The fourth workshop took place at VTT's VR laboratory, where the third and fourth steps of the four-step method were tested. The focus of the VE system was on testing commercial visualisation and/or VE software. The fifth workshop was hosted at VTT's VE laboratory with a new VE application. The platform was based on a PC with Virtools and included an active stereo system, hand and head tracking, data gloves and game controls (See Figure 26). HMD was also tested. The VE application was based on a fork lift model and working process was picking up objects in a warehouse. The main focus of the workshop was on testing the PC system for end-user needs.

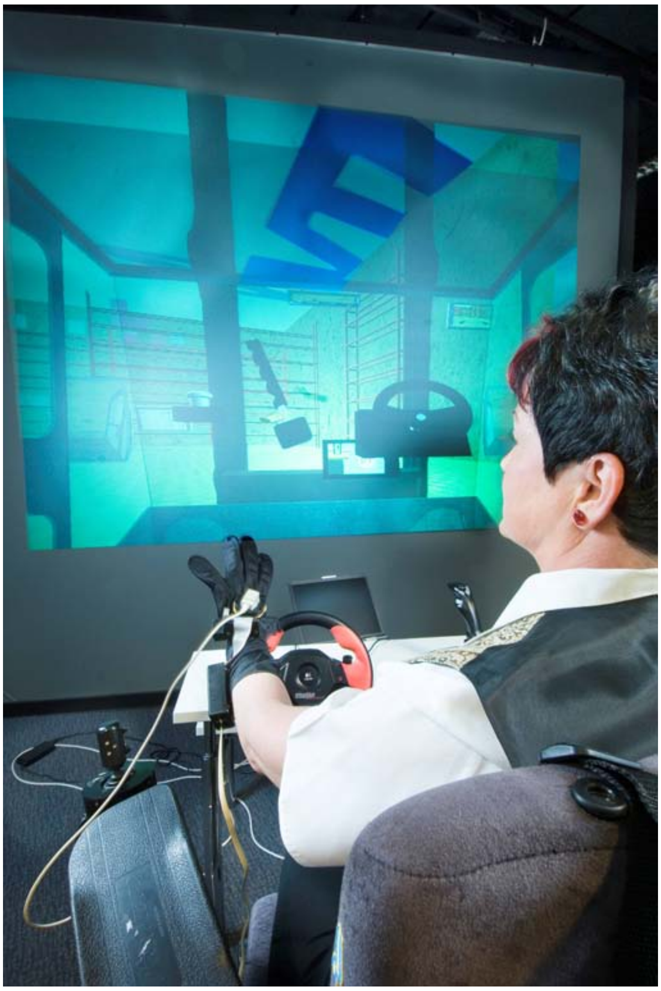


Figure 26. Example of designing cabin layout in VR with data glove.

6.2 Case Study Results

The four-step user-centred method was tested in five industrial case workshops. The majority of the results were based on data that was collected during those workshops. The first workshop was intended for designers and researchers from different organisations. The main goal of the workshop was to commit the movable machines and vehicles designers to using the new user-centred design method and also to using the new design instrument, i.e. VEs. So, different VR applications were introduced to the designers in the first workshop. Both one-wall (3x3 meters) passive stereo and 19" active stereo monitor display systems were used. The designers felt that the 19" monitor could not give an accurate feeling of distances in VEs due to the fact that the viewing angle was too small. They also felt that with the 19" monitor, the model looked more like a child's toy than a real 3D model of a movable machine (the case studies are described in Section 6.1).

The second workshop focused on producing work processes for tillage work. The case study used in this workshop was a tractor (see Figure 25). The main result was that the designers and end-users recognised the importance of a detailed work process structure in order to increase the understanding of the needs of a real working machine, which, in this case, was a tractor. In other words, the designers of the machines can have more knowledge about the real conditions of machinery use by using VR applications.

Interaction between the user and the tractor was via a control system (see Section 2.5.1). The users of the control system found no significant difference between the gaming devices (joysticks, etc.) and the real tractor's control devices in the concept design phase. This may be due to the fact that the model itself is also a rough sketch from a real machine and there is no need for a detailed user interface in this design process phase. However, it is important that the user can execute certain interaction operations with a device that works in the same way as it would in the real world. Another finding from the second workshop was that the use of digital human models with CAD software can be more informative than the use of a detailed VR application in the concept design phase, such as during a coverage area test or when the user's reach estimation is made.

A CAVE-like system was used in the third workshop. The system was a 5-wall active stereo system with a space-mouse interaction device. In this environment the machine designers really felt they were inside a virtual model. They also felt that it was important to have the possibility to see their own hands during interaction with virtual models in the VE. The designers felt confident about the fact they could produce a reliable visibility test in a CAVE-like system, thanks to the wide angle of view. The CAVE-like environment was used in this industrial case workshop for demonstrating state-of-the-art VR equipment for use in a user-centred design process.

The fourth workshop was intended to analyse the four-step design method with designers and researchers. The result from this workshop was that the steps and sub-methods of the four-step design method should be implemented on computers and linked to a current test situation based on design phase-oriented guidelines and a procedure that affords proper sub-steps. This is due to the fact that the sub-methods used were not originally intended for testing with VR applications. The sub-methods need to be modified and computerised in order to consider the demands for VR testing purposes. The machine designers from industrial companies also felt the method and sub-methods should be more compact for industrial use.

The fifth workshop was focused on producing work processes for picking up objects in a warehouse. The case study used in this workshop was a fork lift (see Figure 24). The main result of the workshop was that a PC and Virtools-based VE system is flexible enough for designing cabin layouts. Changing the layout with a data glove was flexible, but controls component library wasn’t large enough. Some kind of tactile feedback was needed. Some companies were planning to exploit the VE system in cabin design.

The main results obtained from the workshops were that systematic use of the four-step user-centred method commits designers to using the design method in all design phases and helps overcome prejudices related to the use of VR equipment and applications. However, there are still deficiencies in the number of suitable commercial VE softwares available; there is still a need for software that is flexible enough for the demands of all design phases.

7. Summary

7.1 Immersive design environment

There are quite a few application areas in which VEs can be utilised. Perhaps the best-known application area is the use of VR applications in training. VR systems also have a long tradition in visualisation applications, like architectural design. In Chapter 2, the focus was on the use of a new user-centred design method and VE applications for user interfaces for cabins.

The first part of Chapter 2 dealt with helping to choose appropriate virtual reality software for use in research and industrial applications. Some basic issues were taken into account in Section 2.2. and a simple-to-use software selection method was briefly introduced in Section 2.4. A simple comparison chart, where all the information was collected was shown and the method guidelines were also explained.

A prototype test environment was designed and implemented as part of the four-step method utilising VTT's VR laboratory facilities. The VR environment – in other words, the test environment – consisted of both hardware and software applications. The test environment setup was introduced in Section 2.5. The test environment and control system was made for movable machines and vehicles design and evaluation purposes. Moreover, one of the system requirements was that the system should also be generic so that it could be used during the project and also later for all kinds of user interfaces for controlling virtual models. The control system was intended for industrial case design workshop test cases.

7.2 The New User-Centred Design Method – The Four-Step Method

In the field of user-centred design, the most important result of the project is the new four-step design method applied to the vehicular working machine design process. The use of user-centred design in the machine design industry could achieve a remarkable competitive edge against competitors. By applying user-centred design together with virtual prototyping during the design process, the most severe design errors in the user interfaces can be repaired before manufacturing. The design method also aims to ensure that the needs of the customers are taken into consideration and that the designers have an in-depth understanding of the work processes the machine should support. The use of the method may also lead to new insights and innovations on how the user's workload can be reduced. This leads to more satisfied customers by providing more value with the product. The developed method was described in detail in the Chapter 3. In brief, it contains the following steps: understanding and specifying the context of use, specifying the user and organisational requirements, producing design solutions and evaluating the designs against the requirements.

7.3 Adaptive user interface

Both adaptive context-aware technologies and multimodal solutions should be considered when developing user interface solutions for sophisticated work machines. An adaptive UI is context-sensitive – i.e. it will adapt to various situations during use. These include changes in the work (the current task status), the user (role, status and way of action), the environment and the status of the work machine. Multimodality introduces the natural way of interaction between human and work machines. Multimodal techniques include, e.g., force feedback, haptics, voice and gestures in addition to traditional WIMP (windows, icons, menus, push buttons) interfaces.

In this section we introduce a context-aware multimodal user interface for vehicularwork machines, describe its implementation in a virtual environment and demonstrate the basic features in work machine applications.

7.4 New Product Development

The basis for generic product development process descriptions has been quite similar since the 1960s, and no longer responds well to the requirements setup in modern companies. The generic process excludes the fact that in many companies the product process (responsibilities of the product-dependent arrangements) continues throughout the lifecycle of the products; maintenance and support often quite directly belong to the responsibilities of product development. However, these sub-processes may also be outsourced. This situation is quite similar when compared to the software development process mentioned in Section 5.2.4.

The success of creating new innovative products and the competence required for frequently introducing breakthroughs onto the market is deeply dependent on the company’s maturity in order to successfully manage the product transfer process presented in Section 5.2.7. The transfer of products through the organisation into the market is of great importance to the profitability of business. Thus the cross-linked functionalities of the product development process, especially the simulation and modelling-based issues (cf. Section 5.2.8), have to be well balanced and focused on relevant matters and have to be well communicated to all involved members and contributors to the product development functions. The earlier the flaws are uncovered, the more features of novelties may be developed further.

7.5 Case Studies

The purpose of the case studies was to evaluate the future development methods described in Chapters 2–5. The case studies were evaluated in industrial workshops by designers from vehicular working machine companies, human factors experts and researchers.

Two real products were used in the case studies: a fork lift and a tractor. Even though the case study products are existing vehicular working machines, the working processes are theoretical. There are five different working processes in the case studies. All the selected processes can be tested with both products in VEs.

Five industrial workshops took place over eighteen months. The main goals of the industrial workshops were to test the method, get feedback from the designers and commit the designers to the use of the new four-step user-centred design method.

The main results obtained from the workshops were that systematic use of the four-step user-centred method commits designers to using the design method in all design phases and helps overcome prejudices related to the use of VR equipment and applications. However, there are still deficiencies in the number of suitable commercial VE softwares available; there is still a need for software that is flexible enough for the demands of all design phases.

8. Conclusions

User-centred design should be applied to the vehicular working machine design process because this would ensure better user interface design in the final products. By using the systematic methods described in this publication, the process could be integrated with current design and manufacturing processes.

Some of the project's industrial partners were interested in integrating the VR system into their design system, but the main difficulties were the lack of standard VR hardware and software, and the lack of detailed guidelines on the VR system's evolution. Another problem was the weak integration of the VR software with the companies' CAD systems.

The change in the content, stages or phases of the generic product development process model since the 1980s has been minor. The progress in the designer's tools, applications and practices, however, is significant. Product development in the real world is taking over other stages and phases of a product's lifecycle, e.g. manufacturing, utilisation, maintenance and support, disposal and recycling. Products are becoming more complex and increasingly include software and other non-mechanical features. A new configurable process model to concurrently manage all aspects of the product's lifecycle and transfer the product onto the market efficiently is innovatively needed.

This project has identified the following future development needs:

- For user-centred design purposes, the VE system should be more interactive and include haptics, feedback and a motion platform.
- Integration with companies' PLM systems.
- Better integration between the new user-centred design method and the flexible VE system.
- The VE models should be more component-based, and include knowledge of all components. This will advance the reuse of the VE models and parameterisation of the VE sub-models.
- This project's main focus was on the user interface design in the cabin, but other issues like ventilation and sound design should also be taken into account in the same design simulator.

This study only partly reached the main objectives – the integration between the four-step user-centred design method and the flexible VE system was too weak – but the methods are applicable to the vehicular working machine design process. The main advance would be to focus on the integration between the enhanced method and the VE system.

Acknowledgements

The study was funded by Tekes (National Technology Agency of Finland) and carried out in the HumanICT research project (decision 40353/04).

References

Andreasen, M.M. & Hein, L. 1987. Integrated Product Development. Bedford:IFS, (Publications) Ltd. 205 p.

Apilo T. 2004. Integrated development project management. IPMA World Congress on Project Management. Budapest, 19–20 June 2004.

Apilo, T. & Taskinen, T. 2006. Innovaatioiden johtaminen [Management of innovations]. Espoo: VTT Tiedotteita – Research Notes 2330. ISBN 951-38-6774-9; 951-38-6775-7. 112 p. + app. 10 s. http://www.vtt.fi/inf/pdf/tiedotteet/2006/T2330.pdf.

Barnes, R.M. 1968. Motion and time study (6th ed.). Wiley, New York.

Beyer, H. & Holtzblatt, K. 1999. Contextual design. Interactions, 6, pp. 32–42.

Bolle, R.M., Connell, J.H., Pankati, S., Ratha, N.K. & Senior, A. W. 2004. Guide to biometrics, Springer-Verlag, New York, USA. 353 p. ISBN 0-387-40089-3.

Brethauer, D.M. 2002. New Product Development and Delivery: Ensuring Successful Products Through Integrated Process Management. Saranac Lake, NY, USA: AMACOM.

Bullinger, H.J. & Dangelmaier, M. 2003. Virtual prototyping and testing of in-vehicle interfaces. Ergonomics, 46, pp. 41–51.

Burnett, G.E., Summerskill, S.J. & Porter, J.M. 2004. On-the-move destination entry for vehicle navigation systems: Unsafe by any means? Behaviour & Information Technology, 23, pp. 265–272.

Cagan, J., Nussbaum, B. & Vogel, C.M. 2002. Creating Breakthrough Products: Innovation from Product Planning to Program Approval. Prentice Hall. 336 p.

Chaffin, D. B. 1975. Ergonomics guide for the assessment of human static strength. American Industrial Hygiene Association Journal, 36, pp. 505–511.

Cobb, S., D’Cruz, M. & Wilson, J. 1995. Integrated manufacture: A role for Virtual reality? International Journal of Industrial Ergonomics, 16, pp. 411–425.

Cowan, N. 2000. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. Behavioral and brain sciences, 24, pp. 87–185.

Dey, A.K. 2000. Providing Architectural Support for Building Context-Aware Applications, Ph. D dissertation, Georgia Institute of Technology.

Engström, J., Johansson, E. & Östlund, J. 2005. Effects of visual and cognitive load in real and simulated motorway driving. Transportation Research Part F, 8, pp. 97–120.

Green, P. 1999. Estimating Compliance with the 15-Second Rule for Driver-Interface Usability and Safety. Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting. Santa Monica, CA: Human Factors and Ergonomics Society.

Green, S., Gavin, M. & Aiman-Smith, L. 1995. Assessing a multidimensional measure of radical technological innovation, IEEE Trans. Eng. Mgt., 42(3), pp. 203–214.

Hackos, J.T. & Redish, J.C. 1998. User and task analysis for interface design. John Wiley & Sons.

Hansen, F. 1965. Konstruktionssystematik. Berlin: VEB-Verlag Technic.

Heikkilä, R. & Jaakkola, M. 2002. The Efficiency of a 3-D Blade Control System in the Construction of Structure Layers by Road Grader – Automated Design-Build of Road Construction in Finland, ISARC 2002, Proceedings of 19th International Symposium on Automation and Robotics in Construction, Washington DC, USA, 23–25 September, 2002. Pp. 475–480.

Hirchheim, R. 1989. User participation in practice: experiences with participative system design. In: Knight, K. (ed.). Participation in systems development, Columbia GP Publishing. Pp. 194–204.

Hubka, V. 1987. Principles of engineering design. Zürich: Heurista. 118 p.

ISO 13407. 1999. Human-centered design processes for interactive systems. International Organization for Standardization, Geneva.

ISO 9241-11. 1998. Guidance on Usability. International Organization for Standardization, Geneva.

Jain, A. K. & Ratha, N.K. (eds.). 2005. Biometric Technology for Human Identification II. Proceedings of SPIE, Vol. 5779, SPIE, March 2005. Pp. 484. ISBN 0-819-45764-7.

Jayaram, S., Connacher, H. & Lyons, K. 1997. Virtual assembly using virtual reality techniques. Computer-Aided Design, Vol. 29, 8, pp. 575–584.

Juran, J.M. 2001. Process Management. Berkeley, CA, USA: Osborne/McGraw-Hill.

Kalawsky, R. 1993. The Science of Virtual Reality and Virtual Environments. Addison-Wesley Pubishing Company.

Kallela, J. 1996. Automaation paradigmat. Käyttäjien osallistuminen automaation suunnitteluun. [Paradigms in automation. User participation to automation design]. Technical Research Centre of Finland. Espoo: VTT Julkaisuja – Publikationer 817. 148 p. + app. 74 p. ISBN 951-38-4526-5.

Kantowitz, B.H. & Sorkin, R.D. 1983. Human Factors: Understanding People-System Relationships. Wiley.

Karhu, O., Kansi, P. & Kuorinka, I. 1977. Correcting working postures in industry: A practical method for analysis. Applied Ergonomics, 8, pp. 199–201.

Kauppi, I. 2003. Intermediate Language for Mobile Robots. A link between the high-level planner and low-level services in robots. Espoo: VTT Publications 510. 143 p. ISBN 951-38-6251-8; 951-38-6252-6.
http://www.vtt.fi/inf/pdf/publications/2003/P510.pdf.

Kilpeläinen, P., Nevala, K. & Heikkilä, R. 2004a. Development of a control system for road construction automation applications, Proceedings of 21st International Symposium on Automation and Robotics in Construction, IAARC, Jeju, South Korea, 21–25 September 2004. Pp. 65–70.

Kilpeläinen, P., Nevala, K., Tukeva, P., Rannanjärvi, L., Näyhä, T. & Parkkila, T. 2004b. Älykäs tietyömaa. Tienrakennuskoneiden modulaarinen ohjaus [The intelligent construction site. The modular control system for the road construction machines]. Espoo: VTT Tiedotteita – Research Notes 2255. 116 p. ISBN 951-38-6486-3; 951-38-6487-1.
http://www.vtt.fi/inf/pdf/tiedotteet/2004/T2255.pdf.

Lee, H.C., Lee, A.H., Cameron, D. & Li-Tsang, C. 2003. Using a driving simulator to identify older drivers at inflated risk of motor vehicle crashes. Journal of Safety Research, 34, pp. 453–459.

Ligier, A., Fliedner, J., Kajanen, J. & Peyret, F. 2001. Open system for road information support, Proceedings of 18th International symposium on Automation and Robotics in Construction, Krakow, Poland, 10–12 September, 2001.

Määttä, T. 2003. Virtual environments in machinery safety analysis. PhD Thesis. Espoo: VTT Publications 516. 170 p. + app. 16 p. ISBN 951-38-6261-5; 951-38-6262-3. http://www.vtt.fi/inf/pdf/publications/2003/P516.pdf.

Määttä, T., Naumanen, M. & Karhu, R. 2005. User interface in future working machines – views from a technology roadmap. In: Kleemola, H. (ed.). Industrial Systems Review 2005, Technical Research Centre of Finland, Tampere. Pp. 12–17.

Mäntyjärvi, J. 2003. Senror-based context recognition for mobile applications. Espoo: VTT Publications 511. 118 p. + app. 60 p. ISBN 951-38-6253-4; 951-38-6254-2. http://www.vtt.fi/inf/pdf/publications/2003/P511.pdf.

Määttä, T., Viitaniemi, J., Leino, S.-P., Helin, K. & Aromaa, S. 2003. Virtual environment based procedure for production task evaluation and design. International Federation of Automatic, Control 8th IFAC Symposium. Göteborg, SE, 2003, Automated Systems Based on Human Skill and Knowledge, Chalmers University of Technology, Göteborg. Pp. 203–208.

Muller, M.J. & Kuhn, S. 1993. Participatory design. Communications of the ACM, 36, pp. 24–28.

Mumford, E. 1989. User participation in a changing environment – Why we need it? In: Knight, K. (ed.). Participation in system development, GP Publishing, Columbia. Pp. 60–72.

Nielsen, J. 1993. Usability engineering. Academic Press.

Nishiguchi, T. (ed.). 1996. Managing Product Development. Cary, NC, USA: Oxford University Press, Inc.

Norman, D.A. 2005. Human-centered design considered harmful. Interactions, 12, pp. 14–19.

Norman, D.A. 2002. The design of everyday things. The MIT Press.

Norros, L., Kaasinen, E., Plomp, J. & Rämä, P. 2003. Human-Technology Interaction. Research and Design. VTT Roadmap. Espoo: VTT Tiedotteita – Research Notes 2220. 118 p. + app. 11 p. ISBN 951-38-6196-1; 951-38-6197-X. http://www.vtt.fi/inf/pdf/tiedotteet/2003/T2220.pdf.

Pahl, G. & Beitz, W. 1995. Engineering Design. Springer-Verlag, London.

Pahl, G. & Beitz, W. 1977. Konstructionslehre: Handbuch für Dtudium und Praxis. Berlin: Springer.

Pahl, G. & Beitz, W. 2003. Engineering Design. A systematic approach. 2nd edition. London: Springer-Verlag. 544 p.

Pease, R.W. Jr. 2002. Merriam-Webster's medical dictionary. Merriam-Webster Inc.

Pugh, S. 1996. Creating Innovative Products Using Total Design. Addison Wesley. 600 p.

Pugh, S. 1991. Total Design, Integrated Methods for Successful Product Engineering. Essex: Pearson Education Limited. 278 p.

Rajlich, V.T. & Bennett, K.H. 2000. A Staged Model for the Software Life Cycle. Computer, 33(7), pp. 66–71.

Rasmussen, J. 1986. Information processing and the human-machine interaction: an approach to cognitive engineering. New York, North Holland.

Reason, J. 1990. Human error. Cambridge University Press.

Roozenburg, N. & Cross, N. 1991. Models of the design process – Integrating across the disciplines. In: Hubka, V. (ed.). Proceedings of the International Conference on Engineering Design in Zürich, Vol. I. Zürich, Heurista. Pp. 186–193.

Roth, K. 1981. Konstruieren mit Konstruktionskatalogen. Berlin: Springer-Verlag. 475 p.

Saariluoma, P. 2004. Explanatory frameworks in interaction design. In: Pirhonen, A., Isomäki H., Roast C. & Saariluoma P. (ed.). Future Interaction Design, pp. 67–83.

Sääski, J. 2006. Simulation-based design process. In: Lehtonen, M. (ed.). Simulation-based design process of smart machines. Espoo: VTT Tiedotteita – Research Notes 2349. Pp. 12–38. ISBN 951-38-6809-5; 951-38-6810-9.
http://www.vtt.fi/inf/pdf/tiedotteet/2006/T2349.pdf .

Santos, J., Merat, N., Mouta, S., Brookhuis, K. & de Waard, D. 2005. The interaction between driving and in-vehicle information systems: Comparison of results from laboratory, simulator and real-world studies. Transportation Research, Part F, 8, pp. 135–146.

Savioja, P. 2003. Käyttäjäkeskeiset menetelmät monimutkaisten järjestelmien vaatimusten kuvaamisessa [User-centred methods in presenting the requirements of complex systems]. Espoo: VTT Tiedotteita – Research Notes 2216. 132 p. + app. 10 p. ISBN 951-38-6186-4; 951-38-6187-2. http://www.vtt.fi/inf/pdf/tiedotteet/2003/T2216.pdf.

Schilit, B., Adams, N. & Want, R. 1994. Context-Aware Computing Applications, Proceedings of the Workshop on Mobile Computing Systems and Applications, IEEE Computer Society, Santa Cruz, USA, December, 1994. Pp. 85–90.

Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., Van Laerhoven, K. & Van de Velde, W. 1999. Advanced Interaction in Context. In: H. Gellersen (ed.). Handheld and Ubiquitous Computing, Lecture Notes in Computer Science No. 1707, Springer-Verlag, Heidelberg. Pp. 89–101. ISBN 3-540-66550-1.

Smith, R.C. 2001. Shared vision. Communications of the ACM, 44, pp. 45–48.

Stone, R. 2001. Virtual reality for interactive training: an industrial practitioner's viewpoint. Int. J. Human-Computer Studies, 55, pp. 699–711.

Suh, N.P. 1990. The Principles of Design. Oxford, UK: Oxford University Press.

Sundin, A. 2001. Participatory Ergonomics in Product Development and Workplace Design. Supported by Computerised Visualisation and Human Modelling. PhD Thesis. Gottenberg, Chalmers University of Technology.

Teakle, P.R. & Gilmore, D.B. 1999. New product development. In: Kreith F. (ed). Mechanical Engineering Handbook. CRC Press LLC, Boca Raton.

Tinsley, T., Belytschko, T., Fish, J., Hughes, T., Johnson, C., Keyes, D., Laub, A., Petzold, L., Srolovitz, D., Yip, S. & Bass, J. 2006. Simulation-Based Engineering Science. Revolutionizing Engineering Science through Simulation. National Science Foundation, Arlington, VA USA, Report No.(s): PB2006-114248. 92 p.

Tjalve, E. 1979. A short course in industrial design. U.K., London: Butterworth & Co. 207 p.

Tromp, J.G. & Nichols, S.C. 2003. VIEW-IT: A VR/CAD inspection tool for use in Industry, Proceedings of the HCI International 2003 Conference, Crete, 2003.

Ulrich, K.T. & Eppinger, S.D. 2002. Product Design and Development. 2nd ed. NewYork: McGraw-Hill Inc.

Ulrich, K.T. & Eppinger, S.D. 1995. Product Design and Development. NewYork: McGraw-Hill Inc.

VDI 2221. 1987. Systematic Approach to the Design of Technical Systems and Products. Düsseldorf: VDI Society for Product Development, Design and Marketing. 34 p.

Vicente, K.J. 1999. Cognitive Work Analysis. Toward Safe, Productive, and Healthy Computer-Based Work. Lawrence Erlbaum.

Viitaniemi, J., Helin, K., Aromaa, S., Montonen, J., Määttä, T. & Salmi, T. 2006. A simulation tool for production and human activity design. CIRP ICME '06, 5th CIRP International Seminar on ICME Intelligent Computation in Manufacturing Engineering, Ischia, Italy, 2006.

Viitaniemi, J., Leino, S.-P., Helin, K. & Määttä, T. 2001. Participating approach to work task design applying virtual environment, Proceedings of NES 2001. 33rd Annual Congress of the Nordic Ergonomics Society. Tampere, FI, 2–5 Sept. 2001. University of Tampere. Pp. 74–77.

Wang, Z., Liu, Z., Tan, J., Fu, Y. & Wan, C. 2006. A virtual environment simulator for mechanical system dynamics with online interactive control. Advances in Engineering Software, 37, pp. 631–642.

Wierwille, W.W. 1993. Visual and manual demands of in-car controls and displays. In: Peacock, B. & Karwowski W. (ed.). Automotive ergonomics, Taylor & Francis. Pp. 299–320.

Wilson, J.R., D'Cruz, M., Cobb, S. & Eastgate, R. Virtual reality for industrial applications. Nottingham University Press, 1996.

Wilson, J.R., Eastgate, R.M. & D'Cruz, M. 2001. Structured Development of Virtual Environments. In: Stanney, K. (ed.). Handbook of Virtual Environments. Lawrence Erlbaum Associates. Pp. 353–378.

Woodson, W.E. 1981. Human factors design handbook. McGraw-Hill, New York. 2nd Ed. John Wiley and Sons, New York.

Zelzer, D. 1992. Autonomy, interaction and presence. Presence, 1, pp. 127–132.

Zelzer, D. 1990. Virtual environments: Where are we going? Proceedings of IDATE 12th Annual Conference. Pp. 507–513.

Appendix A: A basic example of a comparison chart

	General						
	Software ABCD (www.abcd.com)	**Price**	**Maintenance cost**	**Manufacturer**	**Technical support & location**	**Demo version**	**Licensing Politics**
Theoretical comparison	**Software ABCD** (www.abcd.com)	Licensing: - Basic license 50ke (for one computer) -Basic license 50ke+5ke / extra computer (floating license) Extra modules: - Digital Human Model 8ke - VR-interfaces 6ke	Annual Licensing costs: - 20% of purchasing price Includes: - technical support - software upgrades	ABCD is an international company, and it is a part of a bigger consortium formed by well-known CAD-software company. The head-office is located in the USA. Over 30 agensies known world wide. The ABCD company is founded in 1993, it has 120 employees.	Nearest support office: - Sweden, Stockholm Support by: - e-mail - web - phone - implementation services can be included Cost: - Included in annual licensing fees	Can be downloaded from Internet-pages. It is a version for 30 days. They will give all the demo-packages needed. Technical support and manuals included.	Possibility to borrow a floa license, e.g for travel purpoces
	Ranking	**1**	**3**	**2**	**1**	**1**	**2**
Technical issues			**VR-interfaces**			**Other issues**	
Operating system	**Cluster of computers**	**Training**	**Tracking**	**Sound**	**Control devices**	**Other interfaces**	**References**
Win 2000 / XP (32- ja 64-bit)	Availability of a cluster solution. Cost 5 ke / node.	Training is 1000e / day for 5 persons. Min. is a 3 days training period. Possibility to take a part in training in Sweden or USA, then the price is 500 / day.	Supported systems: - Ascension Flock of Birds - Polhemus Fastrak (3rd party license needed, 5000e)	Dolby digital 5.1	- Space mouse - all USB-controls - force feedback not available - 5DT data glove	One can program interfaces, for e.g.: - dynamic engine - more control devices	Over 11.000 liceses sold \ wide Companies, already uses software A,B and C More informa available -www.xyz.fi - journal of… - Technical Research ins of..
2	**1**	**1**	**2**	**3**	**1**	**1**	**1**

VTT Working papers

52 Alanen, Raili & Hätönen, Hannu. Sähkön laadun ja jakelun luotettavuuden hallinta. State of art -selvitys. 2006. 84 s.

53 Pasonen, Markku & Hakkarainen, Toni. Kaukolämpölinjojen elinikä ja NDT. 2006. 27 s.

54 Hietaniemi, Jukka, Toratti, Tomi, Schnabl, Simon & Turk, Goran. Application of reliability analysis and fire simulation to probabilistic assessment of fire endurance of wooden structures. 2006. 97 p. + app. 23 p.

55 Holttinen, Hannele. Tuulivoiman tuotantotilastot. Vuosiraportti 2005. 2006. 38 s. + liitt. 7 s.

56 Häkkinen, Kai, Hemilä, Jukka, Salmela, Erno & Happonen, Ari. Logistiikka Belgiassa. Vierailukokemuksia keväältä 2006. 2006. 33 s.

57 Kulmala, Risto. Tiehallinto ja liikenteen tietopalvelut. Selvitysmiehen muistio. 2006. 29 s. + liitt. 3 s.

59 Graczykowski, Cezary & Heinonen, Jaakko. Adaptive Inflatable Structures for protecting wind turbines against ship collisions. 2006. 86 p. + app. 39 p.

60 Känsäkoski, Markku, Kurkinen, Marika, von Weymarn, Niklas, Niemelä, Pentti, Neubauer, Peter, Juuso, Esko, Eerikäinen, Tero, Turunen, Seppo, Aho, Sirkka & Suhonen, Pirkko. Process analytical technology (PAT) needs and applications in the bioprocess industry. Review. 2006. 99 p.

61 Välisalo, Tero, Räikkönen, Minna & Lehtinen, Erkki. Asset Management vesihuollossa. Kirjallisuustutkimus. 2006. 79 s. + liitt. 8 s.

62 Holt, Erika. Current trends in USA building research. 2006. 26 p. + app. 7 p.

63 Karanta, Ilkka. Methods and problems of software reliability estimation. 2006. 57 p.

64 Forsström, Juha. Ydinjätehuollon kustannusriskianalyysi. Esitutkimus. 2006. 51 p.

65 Talja, Asko, Törnqvist, Jouko, Kivikoski, Harri, Carpén, Leena & Nippala, Eero. Ruostumaton teräs maa- ja vesirakentamisessa. 2006. 32 s. + liitt. 5 s.

66 Rinne, Tuomo, Hietaniemi, Jukka & Hostikka, Simo. Experimental Validation of the FDS Simulations of Smoke and Toxic Gas Concentrations. 2007. 37 p. + app. 9 p.

67 Fox, Stephen. Three Original Methods for Minimizing Ambiguities in Global Project Communications. 2007. 50 p.

68 Eckhardt, Jenni, Öörni, Risto, Hautala, Raine, Lehtonen, Mikko & Leviäkangas, Pekka. Tietopalvelumalli. Yleinen malli tietopalvelujärjestelmien kuvaamiseen ja arviointiin. 2007. 22 s. + liitt. 6 s.

69 Kanniainen, Jenni & Haajanen, Jyrki. BPEL Engines. State-of-the-Art Survey for SOAMeS-project. 2007. 30 p.

70 Hemilä, Jukka, Happonen, Ari & Jansson, Kim. Vendor Managed Inventory models in Sweden. Industrial benchmarking experiences from autumn 2006. 2007. 26 p.

71 Hietaniemi, Jukka & Rinne, Tuomo. Historiallisesti arvokkaan kohteen toiminnallinen paloturvallisuussuunnittelu. Esimerkkitapauksena Porvoon museo. 2007. 136 s. + liitt. 44 s.

72 Sippula, Olli, Hokkinen, Jouni, Puustinen, Harri, Yli-Pirilä, Pasi & Jokiniemi, Jorma. Fine Particle Emissions from Biomass and Heavy Fuel Oil Combustion without Effective Filtration (BIOPOR). 2007. 37 p. + app. 4 p.

73 Helin, Kaj, Evilä, Teemu, Viitaniemi, Juhani, Aromaa, Susanna, Kujala, Tuomo, Patel, Harshada, Pakkanen, Toni, Raisamo, Roope, Salmenperä, Pekka, Miettinen, Juha, Kilpeläinen, Pekka, Rannanjärvi, Leila & Vähä, Pentti. HumanICT. New Human-Centred Design Method and Virtual Environments in the Design of Vehicular Working Machine Interfaces. 2007. 85 p. + app. 1 p.

ISBN 978-951-38-6625-9 (URL: http://www.vtt.fi/publications/index.jsp)
ISSN 1459-7683 (URL: http://www.vtt.fi/publications/index.jsp)