Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Personalized Use Models in Ambient Intelligence Environments

Daniel Görlich*, Nancy Thiels, Gerrit Meixner****

**University of Kaiserslautern, Kaiserslautern, Germany*
(e-mail: goerlich@mv.uni-kl.de)
*** German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany*
(e-mail: [nancy.thiels|gerrit.meixner]@dfki.de)

Abstract: The impact of user interface quality has grown in software systems engineering, and will grow further with upcoming new paradigms such as Ambient Intelligence (AmI) or Ubiquitous Computing, which confront the production industry with a huge diversity of new usage situations. User in such situations cannot cope with the vast amount of information. Therefore, new models with respect to users' needs are required to support them fulfilling their tasks. In this paper, the adaptation of a task-oriented use model to future paradigms is presented by considering personal structural preferences and advanced user interaction control. The model reflects user groups' tasks and user interface structure preferences described in a system-independent language. For the future, this model is intended to be used for the run-time generation of user interfaces for adaptive software and intelligent environments, especially in the area of production and manufacturing.

1. INTRODUCTION

The level of acceptance of a user interface depends largely on its ease and convenience of use. A user can work with a technical device more efficiently when the user interface is tailored to the users' needs, on the one hand, and to their abilities on the other hand. Therefore, during a systematic development process, the users' needs, preferences, tasks, and mental models have to be surveyed, in order to subsequently deploy them into the development of a task-oriented and user-friendly device that will be as convenient as possible to use.

Still, people think and act quite differently, even when they perform the same task. Their personal requirements may depend on a large variety of influences ranging from their qualification, their area of activity and their tasks, up to rapidly changing conditions such as mood, time of day, current location, or recent events.

The ongoing technological development of microelectronics and communication technology does not only lead to more pervasive communication between single peers like digital photo cameras and color printers, but will further result in entire pervasive networks of "everyday" devices, as elaborated in (Remagnino and Foresti, 2005). As of today, already 98% of all CPUs are embedded, for example, in modern cellular phones or nearly all kinds of home appliances. Furthermore, distributed computing power – also for industrial devices and components – is continuously rising.

Communication, connectivity and networking can be perceived as services that ultimately always serve the interest of humans in terms of gathering and consolidating distant information. To this date, networked devices are mainly found in consumer goods intended to facilitate and enrich everyday life at home as well as at work. This collaboration of consumer goods is being researched, developed, evaluated, and finally demonstrated in several so-called "smart homes," centers of excellence regarding the technologically advanced ways of life.

In production environments, it is nowadays common to train employees in the operation of devices and restrict their access to safety-critical device functions, so all users (operators) are taught how to handle a device in advance. In a private environment, e.g., at home, however, users often have nothing more than a manual describing the functionality of a certain device – and they often refuse to read it right away.

The personalization of user interfaces has entered many fields – particularly the consumer product industry. The advantages of personalization concepts become clear when using such systems: offered information is adjusted to the needs or previous use habits of users. In contrast to the consumer product industry, the personalization of user interfaces and the orientation on users' needs are still rare in the production environment. Developers have a certain understanding of user needs and interests related to user interfaces. This understanding – no matter, how far from reality or whether at all it is applicable – is realized in the development of user interfaces. Actual needs of users remain unconsidered (Zühlke, 2004).

One approach to overcome this lack relies on the identification of user groups. Studies in the field of user-group-specific prototypes have shown that distinctive advantages result from these user interfaces; higher efficiency and a faster learnability can be achieved by structuring and designing prototypes for certain specific user groups (Wittenberg, 2004). For developing user interfaces, the structure of tasks and functions are the basis for the further user interface design process and consequently for the usability of the user interface itself. Therefore it is important to know the structural preferences of those users working with the interface. By considering these structural preferences, users are enabled to interact quickly and intuitively when doing their work.

In this paper, the influence of structuring preferences on a use model for the development of user interfaces is being described. Structural preferences are given and the adapted model is being explained.

The remainder of this paper is structured as follows: Section 2 gives an overview of the current Useware development process. In Section 3, we describe aspects of the actual use model. Section 4 deals with personalization aspects. Subsequently, Section 5 describes the adaptation of the Useware development process and the use model to the new Ambient Intelligence paradigm. Finally, in Section 6 we conclude and explicate our future work.

2. THE MODEL-DRIVEN USEWARE DEVELOPMENT PROCESS

The levels of acceptance and efficiency of a modern user interface are strongly determined by their ease of use. System development has been advanced by the Center for Human-Machine-Interaction (ZMMI) at the German Research Center for Artificial Intelligence (DFKI) by the development of a comprehensive, systematic Useware development process (Zühlke, 2004). The primary considerations in this evolutionary process are always the requirements and needs of the users, for whom the user interface is being developed. This is the only guarantee for an efficient use of the system.

A model-driven approach requires that models are defined and generated at certain stages during the development process. Therefore, there must be a specific flow of development activities. When developing user interfaces for machines, the Useware development process depicted in Figure 1 is applied. It consists of four overlapping phases accompanied by an iterative evaluation phase. The iteration ensures that the results of each step are accessible not only to the developers, but also to the final users.

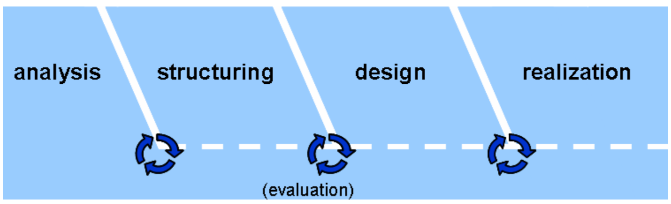

Fig. 1. The model-driven Useware development process

Starting with the *analysis* phase, data about user tasks, their mental models, machine details, the working environment, as well as the organizational structure is colleted. Several data collection methods, e.g. interviews, direct observation of workers in their workspaces, and questionnaires should be applied, since each technique will only provide limited information. The results are documented mainly in a preliminary task model. The collected data provides the basis for the following structuring phase.

Structuring the preliminary task model follows the analysis phase. The previously defined task model and the model of machine functionality are the main input of this phase where only common usage aspects are addressed. This includes defining user groups and their tasks, the usage context, and the accessibility of tasks at different devices and locations. Devices do not play any design role at this stage. Devices can serve as filters that can be used to decide which tasks are available at which device. The resulting use model can be evaluated in terms of logical grouping, decomposition and others. This means, for example, checking whether each task has been placed in the right context and proper decomposition has been done. The model is independent of the later implementation platforms.

Once the use model has been defined, user interface *design* can begin. In the first step, further common design aspects are addressed. Subsequently, a concrete user interface is created by refining abstract aspects by platform-specific ones. User interface prototypes can be directly generated from the use model and tools can be developed exporting the model into required programming languages.

Hard coding (programming) the GUI and implementing it on the target machine is the task of the final *realization* phase. The resulting Useware can then be evaluated regarding design issues, usability, and real time performance.

3. THE USE MODEL

Employing a task model has proven to be a good starting point for user-oriented interface development (Paternò and Santoro, 2002). The preliminary task model is therefore the basis for the use model. Its feasibility to capture user tasks and the way they are performed leads to a focus on the final user during the whole development process (Mukasa and Reuther, 2004).

The use model is defined by using the Useware Markup Language (useML). This is an XML-based markup language for defining and structuring user tasks for machine users). Its main description elements are the use objects (UO) and the elementary use objects (EUO). While the UOs are logically equivalent to sets of related tasks, the EUOs are the elementary actions. A use object therefore expresses a general goal of one or more tasks. Figure 2 shows the useML elements and their relations.

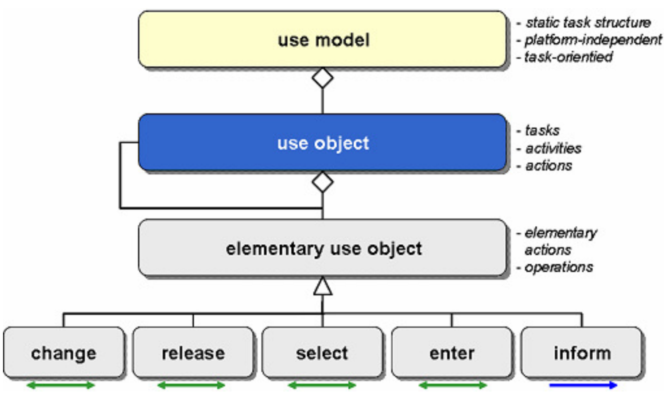

Fig. 2. Current useML scheme

As Figure 2 shows, there are five types of EUOs: *change*, *release, select*, *enter,* and *inform*. The first four ones are bidirectional, i.e. there is an interaction between the user and the machine. *Inform* is unidirectional, i.e. there is no action performed by the user, but the system provides the user with information. These five EUO types correspond to the actions of the machine user and can fully describe all interaction and information needs of users working with technical systems.

Enter involves input of one absolute data value into the machine system. Any previously stored value will be overwritten.

Change, on the other hand, permits relative changes to an existing value or date. It is therefore possible to increment, for example, a speed from 15 $^{m}/_{s}$ to 17 $^{m}/_{s}$ with a pre-defined incrimination factor.

Release implies that the user can directly trigger an action or a machine function resulting in its execution.

Select defines actions where the user can select zero or more values from a set of values that already exist in the system. This selection can lead to changing a parameter in the machine control, for example, changing the unit of speed from $^{km}/_{h}$ to $^{m}/_{s}$, or to triggering a machine function, e.g., changing the machine operation mode from “automatic” to “manual” by selecting the respective mode.

Inform involves the user querying the machine for some information. For example, the user might like to always know the status of the machine. No further interaction is expected here.

With these few but elementary elements, it is possible to define the use model in a platform-independent way as the elements are directly deduced from users’ tasks and extended by commonalities like classifications, priority, etc. For the schema of the use model and useML, please refer to (Reuther, 2003).

The main focus of the current use model is the modeling of single devices or device categories. Requirements of the Ambient Intelligence paradigm were not considered during the developing of the Useware Markup Language.

4. PERSONALIZATION OF USER INTERFACES

The adjustment of user interfaces in the field of individualized software design has already been elaborated (Ritz, 2001). In this area the design of user interfaces depends on the individual user herself and his needs. Applications of these concepts are mainly used in Web interface design and e-Learning (Brücher, 2005). Outrider for this kind of user interface design was, above all, the consumer goods industry. For example, the different possibilities of adjustments to a mobile telephone are various and open a large range of possibilities to the user.

Today, adjustments to the production environment are mostly limited to user groups instead of individual users. User groups can be distinguished by the respective users’ tasks, which means that users of one group share a common set of tasks. Particularly in this environment, an adjustment which allows for specific consideration of individual characteristics and needs is rarely implemented. Even the adjustment to user groups is usually limited to the definition of access restrictions for certain functionalities of respective user interfaces. An advanced adjustment of user interfaces is only possible in rare cases, when the user himself can arrange a part of the surface under restrictions. For example, he could move frequently used functions on top of the screen.

These kinds of adjustments are technically accomplishable for today's operations because only limited access to information and interactions is available to users and exactly one user interface is assigned to one device. Due to ongoing technological advances, the development of individual user interfaces for every usage situation would not be efficient. In the future, however, a better adjustment of user interface design in production environments shall be permitted in order to cope with the huge variety of information and interactions which results from AmI systems. Today’s user interfaces are not appropriate for the use in an AmI environment because AmI systems are characterized by a high level of complexity. Therefore, a dynamic adjustment of user interfaces is needed in order to present the interaction options as well as available and accessible information user-adequately.

In Figure 3, different concepts for the adjustment of user interfaces are presented. It becomes clear that adjustments differ by the kind of technical implementation. The presented concepts can be divided into variable and fixed ones: Variable systems adapt to the inputs of the users, while fixed ones do not respond to different user inputs as for example the change of colours or the arrangement of frequently used functions.

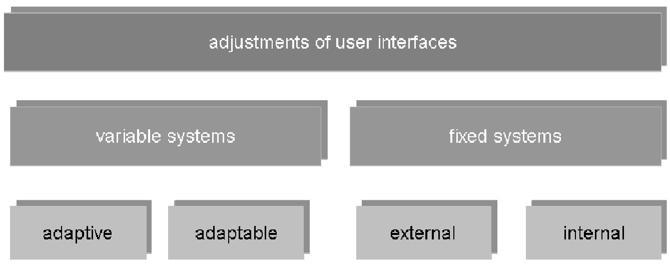

Fig. 3. Adjustment concepts for user interfaces (Thiels et. al., 2006)

Fixed systems respect different aspects of adjustments during their development. At the end of the development process, different adapted systems are generated. For individual users, adapted interfaces result. Within this group, the kind of factors which contribute to the adjustment of user interfaces can be differentiated into internal and external factors. Internal ones are personal attitudes, opinions and desires of users (Bay and Ziefle, 2004). As external factors can be seen occupation, training, and the position of users in their occupational surrounding (Engelbach and van Hoof, 2005).

Variable systems can be divided into adaptive systems, on the one hand, and into adaptable systems, on the other hand (Mertens, *et al*, 2004). Adaptive systems adapt dynamically to user inputs. These systems select the available information and interaction by user inputs. 'Adaptable systems' are systems adaptable by the user in a second step (Hinz, *et al.*, 2004) as he specifies his preferences before or while using the interface.

A dynamic adjustment of user interfaces in terms of adaptability as described above gains importance. However, user interfaces must possess information about the user in order to present exclusively the information necessary for this user and his interactions. Therefore a personal use model was developed as a first step; it will be presented in the next section.

5. PERSONAL USE MODEL

Although the Useware Markup Language is well-suited for the development of single devices or device families, it was not designed to describe more complex production processes or even facilities incorporating a high number of devices or machines of different types. Therefore, the use model must be improved by expanding its scope, providing compatibility for future interaction paradigms such as Ambient Intelligence or Ubiquitous Computing. Such progressive environments will comprise hundreds or even thousands of cooperating devices and embedded systems with which we will quite naturally interact. Traditional interaction paradigms such as GUIs dedicated to a single device may not be sufficient any longer, and users may employ numerous devices at the same time to fulfill their tasks. An appropriate use model therefore must contain a spatial representation of the relevant environments or spaces, as well as a description of devices and device compounds involved in all potential users' works.

In order to meet these new requirements, the use model has been amended with a hierarchical structure of (mobile or stationary) organizational rooms, which may exist physically, or may identify purely logical rooms. In these rooms, device compounds consisting of subordinate device compounds or devices may be located, with which a user can interact in different ways depending on his location and distance. The potentially differing task and use models resulting from the user's and the device's location are annotated to interaction zones. These interaction zones can move, for example, when a mobile device is used in a stationary or even a mobile environment. For a complete analysis of these additions refer to (Görlich and Breiner, 2007). Yet, this representation allows for complex interactions between users and numerous devices in an Ambient Intelligence environment being described in a single, though complex model. It may comprise different use models depending on the users' personal preferences for hierarchical or network structures. Additional personal preferences regarding the design of user interfaces particularly affect the design phase, but not yet the structuring at all. Please see (Thiels and Zühlke, 2007) for further details on this subject!

Further, the use model was expanded by the type of the structure of the latter user interface in order to be able to consider structural user preferences. User tests were performed to analyse the different structural preferences (Thiels et al., 2007). As an outcome of these tests, two different concepts for technical user interfaces exist: a hierarchical and a network structure. Thereby 72% of the tested persons preferred a hierarchical structuring and 16% a network structure. A clear structuring preference of the other tested persons was not possible to evaluate.

But only one of these two preferences can describe a personal use model. User preferences therefore directly influence the use model by identifying structural preferences, specified by the attribute 'structure' in the use model (see Figure 4).

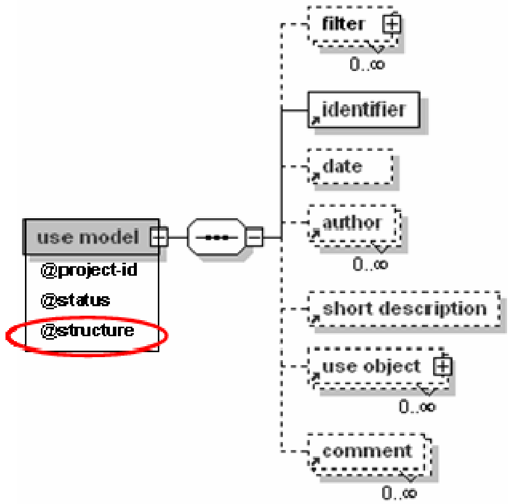

Fig. 4. Extension of the XML scheme for the use model

Particular values of the attribute 'structure' determine the kind of structure, which forms the basis for the future user interface. Furthermore, this value restricts how and/or whether at all use objects can be linked. That can lead to two different use models that result from the user specific preference. These use models differ therefore by the attribute 'structure' (as indicated by the red circle in Figure 4).

Additionally, UOs and several EUOs received a further element, which describes their linkage to other use objects and/or to elementary use objects on basis of the task models from the analysis phase (see Figure 6). The linkage contains the ID's of the target objects to which the current object can be linked because of their content-based connection. The change in the pattern of the exemplary 'release' EUO is depicted in Figure 5, and that of the use object scheme in Figure 6.

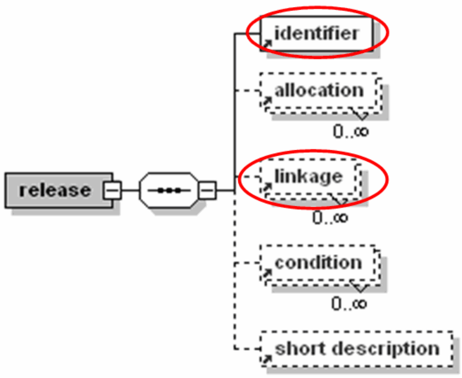

Fig. 5. Extension of the XML scheme for the 'release' EUO

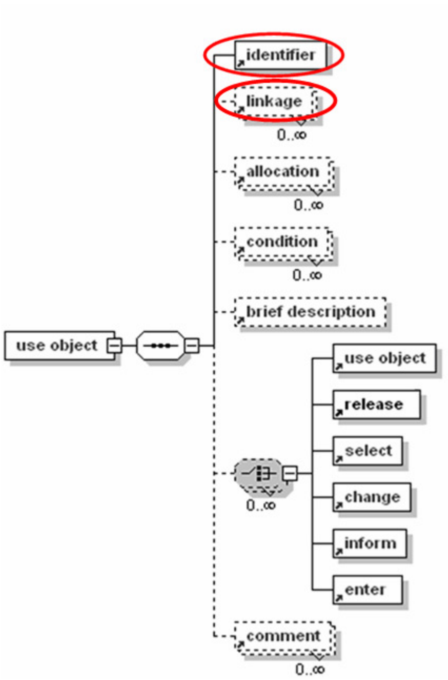

Fig. 6. Extension of the XML scheme for use objects

When several use objects are linked to each other, they span a network as indicated in Figure 7, which shows the updated use model scheme. Further, EUO's can also be interconnected within Elementary Use Object Compounds (see Figure 7) using complex optional temporal relations; the min/max attribute defines how many of these EUO's contained in the compound must or can be executed by the user in which order. Alternatively, the updated use model scheme also allows for specifying activity diagrams.

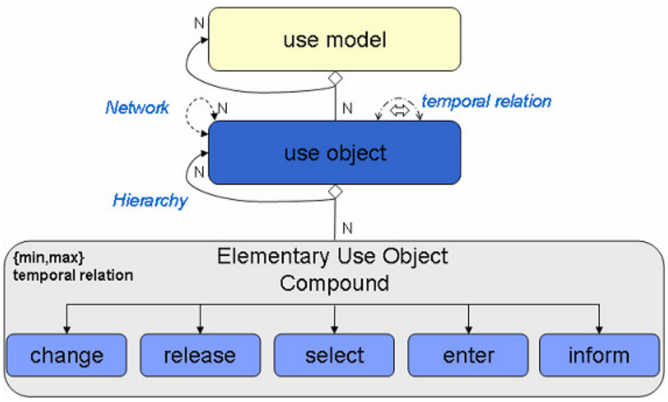

Fig. 7. Extended use model scheme (simplified)

Instances of the use object element 'linkage' are dependent effectively on the value of the attribute 'structure' of the use model: If the attribute 'structure' of the use model contains the value 'hierarchy,' then the linkage elements of the UOs and EUO's are ignored. If it contains the value 'network,' however, then the linkage elements of the UOs and EUOs are considered. Each use object can thereby contain several linkages altogether, but only one to every other specific use object. The consequence of linkage of the UOs and EUOs takes place in the design phase. The content-based linkage will then be transferred to buttons, hyper links, keyboard function keys or short cuts, depending on the content and the purpose.

6. CONCLUSION AND FUTURE WORK

This paper presented our research results aiming to develop an approach for Model-based User Interface Development in Ambient Intelligence Environments. We introduced the actual Useware development process which is supported by the actual use model. Emphasis was put on the adapted use model with respect to future paradigms like Ambient Intelligence and Ubiquitous Computing. For the future, these new concepts must be enhanced and evaluated. The evaluation will take place by generating personal user interfaces, and user tests will be performed to analyse the effects on the usage of personal interfaces.

Efforts must be done developing an adapted Useware Engineering framework and a software tool-chain which can cope with the requirements of these new paradigms. Currently we develop a first tool (Meixner and Thiels, 2008) which focuses on supporting the developer in the analysis phase, and we plan to provide useware engineers with a graphical useML editor to simplify developing use models.

REFERENCES

Bay, S. and M. Ziefle (2004). How Instructions Influence Novice Users' Interaction with Mobile Phones. In: *Work on Computing Systems 2004.* (Khalid, H.M., M.G. Helander, A.W. Yeo (Ed.)), pp. 388-393, Damai Science, Kuala Lumpur.

Brücher, H. (2005). *Lokalisierung als Aspekt der Personalisierung mobiler Kommunikationsdienste.* http://www.kbs.uni-hannover.de/~henze/lla02/ abis proceedings.html (18.07.2005).

Engelbach, W. and A. van Hoof (2005). *Aufgaben- und rollenangepasste Konzeption idealer Informationsstrategien.* http://prebis.informatik.uni-leipzig.de/open cms/export/ download/prebis/ABISWorkshop_10_02 _Engelbach.pdf (18.07.05)

Görlich, D. and K. Breiner (2007). Intelligent task-oriented user interfaces in production environments. In: *1st Workshop on Model-Driven User-Centric Design & Engineering (MDUCDE)*, Seoul/Korea, 2007.

Görlich, D. and K. Breiner (2007): Useware modelling for ambient intelligent production environments. Workshop: *Model-Driven Development of Advanced User Interfaces (MDDAUI), MoDELS 2007*, Nashville, 2007.

Hinz, M., Z. Fiala and F. Wehner (2004). Personalization-Based Optimization of Web Interfaces for Mobile Devices. In: *MobileHCI, Lecture Notes in Computer Science.* No. 3160, pp. 204-215.

Mertens, P., M. Stößlein and T. Zeller (2004). *Personalisierung und Benutzermodellierung in der betrieblichen Informationsverarbeitung – Stand und Entwicklungsmöglichkeiten.* Arbeitspapier Nr. 2, Wirtschaftsinformatik I, Universität Erlangen-Nürnberg. http://www.wi1.uni-erlangen.de/veroeffentlichungen/ download/Arbeitsbericht_WI_I_2_2004.pdf (17.08.2005)

Mukasa, K. and A. Reuther (2004). The Useware Markup Language (useML) – Development of user-centered Interfaces using XML. In: *Proc. of the 9th IFAC/IFIPS/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*. Atlanta, USA.

Meixner, G. and N. Thiels (2008). Tool Support for Task Analysis. *Workshop "User Interface Description Languages for Next Generation User Interfaces", 26th Annual CHI Conference on Human Factors in Computing Systems.* Florence, Italy.

Paternò, F. and C. Santoro (2002). One Model, many Interfaces. In: *Computer-Aided Design of UI.* Kluwer Academic Publishers Dordrecht, Boston, London.

Remagnino, P. and G. L. Foresti (2005). Ambient Intelligence: A New Multidisciplinary Paradigm. In: *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*. D. E. Brown. Vol. 35, No. 1.

Reuther, A. (2003). *useML – Systematische Entwicklung von Maschinenbediensystemen mit XML.* (Ph.D. thesis) In: *Fortschritt-Berichte pak*. Vol. 8, University of Kaiserslautern.

Ritz, T. (2001). Personalized Information Services – an Electronic Information Commodity and it Production. In: *Electronis publishing'01. 2001 in the digital publishing odyssey: Proceedings of an ICCC/IFIP conference held at the University of Kent at Canterbury.* Kent, UK, 5-7 July 2001, IOS Press, Amsterdam, pp. 48-59.

Thiels, N., M. Ehrmann and D. Zühlke (2006). Model-based development of user-centred control systems in ambient intelligent production environments. In: *Proc. of the 9th IFAC Symposium on Automated Systems Based on Human Skill and Knowledge*. Nancy, France.

Thiels, N. and D. Zühlke (2007). Personalization of user interfaces by elevating individual differences, In: *Proc. of the Ergonomic Society Annual Conference*. Nottingham, UK, pp. 266-271.

Thiels, N., T. Maxeiner and K. Röse (2007). Structural user preferences of interfaces and time orientation. In*: Proc. of the Conference on Human-Computer-Interaction.* Beijing, China.

Wittenberg, C. (2004). Requirements analysis and UI concept development for personalized mobile devices. In: *Proc. of the 9th IFAC/IFIPS/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems.* Atlanta, USA.

Zühlke, D. (2004). *Useware-Engineering für technische Systeme***.** Springer, Berlin.