Developing Applications for Wearable Computers: A Process driven Example

Ernesto Morales Kluge[1], Hendrik Witt[2]

University of Bremen, Germany
[1] BIBA, Bremen Institute of Industrial Technology and Applied Work Science, mer@biba.uni-bremen.de
[2] TZI, Wearable Computing Lab., hwitt@tzi.de

Abstract. Interaction and usability aspects of software systems are critical software quality measures. This paper reports on the development process applied to implement an application to support aircraft maintenance tasks with wearable computers. We describe the different phases of the development as well as tools used by application developers to implement the actual application. Moreover, we provide information on the way user studies and their results were used to support the development process with respect to the chosen user interface.

1 Introduction

Interaction and usability aspects of software systems have become critical software quality measures. Today, more attention is payed to a proper user interface design and its seamless integration into the general software development process. A major challenge here, however, is to support software engineers, that are usually not familiar with the latest HCI knowledge, with tools and processes that allow them to build "good" applications.

When considering applications for wearable computers many challenges, different to traditional desktop applications, arise [1, 2]. For instance, understanding interaction design or the appropriate use of sensor-based context information are two critical factors that decide whether or not wearable computing will ever hit the market in the nearer future. In particular, this is because the usability of an application (which is closely related to performance measures) is getting more important and may even overweight cost measures of industrial applications [3].

One of the first comprehensive reports on the design of a wearable computer including its application is described in [4]. Although wearable computing has not yet reached the market, research has proposed a few demo applications for different domains. For example, Boronowsky el al. [5] developed an application that supports crane maintenance. For this, a combination of a visual interface that can be operated with a data glove device was proposed. In [6] a similar concept was used to support aircraft technicians. Besides this maintenance related application examples, there are also others [7, 8].

In aircraft maintenance there are basically three different scenarios where wearable computing applications can empower maintenance workers [9]: (1) Removal and Installation, (2) Inspection, (3) Trouble Shooting. In this paper we report on the development of an aircraft maintenance application in the removal and installation area. The application was closely developed together with industrial partner within the wearIT@work project [10]. In particular, the application supports the removal and installation procedure of passenger seats in the cabin of an aircraft. The focus will be on the overall process and tools used by application developers to implement the actual application as well as the way user studies where integrated to support the development process and particularly the user interface development.

1.1 Outline

The remainder of this paper is structured as follows: Section 2 provides an overview of the envisioned application and the applied development process. It includes a characterization of its application domain as well as the methods applied for deriving requirements. In section 3 the toolkit that was used to implement the application including its user interface is introduced. The actual implementation of the application is presented in section 4. Section 5 reports on a preliminary evaluation of the developed application. Finally, section 6 concludes the paper and points out some future work.

2 Envisioned Application and Requirements

A specific case will be presented that is located thematically in the field of aircraft maintenance namely in the cabin of an aircraft. A maintenance task was chosen to fit the requirements of the end user and the researching developers.

Through several workplace studies and tests of different I/O modalities at the operators working environment the requirements for the envisioned application were elicited. One key element for the whole development process is the end user and its needs. Another key element is the benefit that should emerge out of the introduction of wearable systems in industrial working environments. By conducting interviews with several maintainers and by observing those while working requirements were identified for the application and its functionalities as well as crucial prerequisites for the introduction of innovative wearable systems. The development process followed this approach (see Figure 1):

Workplace Study and User Requirements (1): The team conducting the interviews consisted of a person responsible for the UI (User Interface) implementation tool used by application developers, the application developer himself, and the scientific support that took also the lead in organizing the development. After having arranged the elicitation of general requirements and working conditions of the maintenance operators, the whole team analyzed and interpreted the collected information and formulated the overall application goal and its

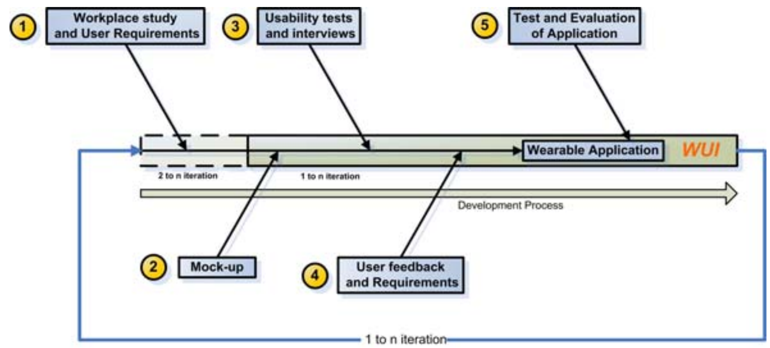

Fig. 1. Application Development Process

functionalities. This analysis led to unique requirements for the wearable application besides requirements concerning, e.g., the morphology and functionalities of equipment that are not in the focus of this paper but are limited in any case to the availability of Commercial of the Shelf (COTS) Components. As a result of the interviews the main constraints of the end users directed to a wearable application can be described as follows:

- Every development (application) must have a benefit for the user. This could be a process improvement e.g. saving of time or reduction of complexity (reduce the number of persons involved in the task).
- The end user stated that the use of wearable equipment is very (only) interesting for the support of complex maintenance procedures that have an inherent danger of confusion e.g. confusing very similar but different parts.

These constraints/requirements are obviously very high level and need to be interpreted with the perspective of an end user. Even when creating a picture of wearable computing during the workplace-studies in the operators' mind it is the task of the development team to transfer high level requirements into innovative and useful innovations or experiments which lead to valid and reliable results.

Building the Application Mock-Up (2): Starting from the point of knowing the users needs and the overall goal mentioned above, the team started developing the application. Due to the nature of the used implementation tool for wearable systems, work started concurrently. I.e. the application developer designed a mock-up with the implementation tool, not knowing how the content will look like in detail, while the scientific part collected the content and the information needed for the real application. The first mock-up showed only the structure of the envisioned system. Because of the iterative approach that was chosen, several mock-ups were developed and with every iteration of the mock-up the range regarding functions and content increased. Accompanied by reviews of the application done by the whole team corrective decisions were taken.

Performing Usability Tests and Experiments (3): As the result of usability tests of input and output modalities, technical requirements emerged. Specifically customized interfaces were tested for *input* modalities: (1) speech, (2) handmouse and (3) dataglove (gesture device). As *output* modalities (1) plain text, (2) image (photo), (3) video and (4) audio were tested. Input and output tests were performed with ten maintenance operators through a mock-up that was created with an implementation tool for wearable systems. Figure 2 show first results of the input modality performance.

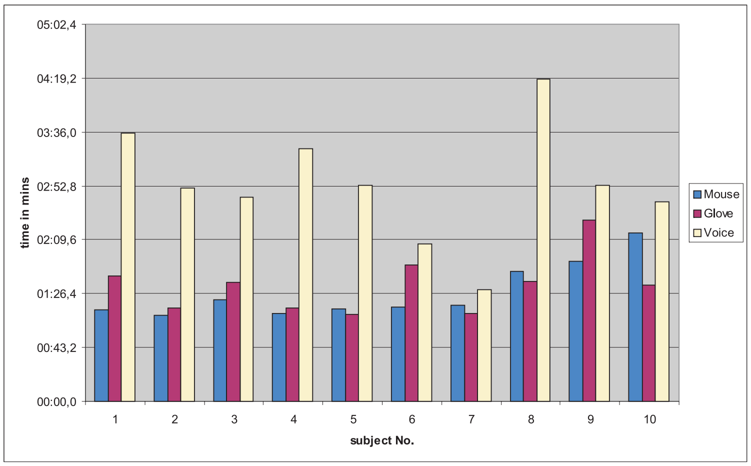

Fig. 2. User Performance with Handmouse, Dataglove and Speech

User Feedback and Requirements (4): The final application of the first application iteration was implemented by using the dataglove (gesture device) and the speech navigation as *input* modality. The team came to this decision basing on the results of the usability tests that showed that the dataglove outperformed compared to the speech navigation and performed nearly equal as the handmouse (as a traditional input device) (Figure 2). Speech was also chosen because interviews of the usability tests showed a significant interest of the subjects in such an innovative technology. Although operators were best with the handmouse, they felt reluctant using the device for interaction with a wearable computer. Operators argued, that holding the mouse device in hand is unpractical and uncomfortable during real work in aeronautic maintenance environments. Instead, they preferred using the dataglove even though they performed worse with it.

The *output* modalities were also chosen after evaluating the results of the usability tests. Results showed that the fastest and most exact modality in respect to correctness was the image output. A combination of the reproduced maintenance manual (text output) and images as well as technical drawings (taken out of the maintenance manual) were selected as output modality (Figure 4).

Test and Evaluation of Application (5): After having passed a not predefined number of mock-up iterations the application was tested together with the end users in a real working environment.

3 The WUI-Toolkit

The Wearable User Interface (WUI) Toolkit [11] is currently developed in the wearIT@work project [10] as an approach to support application developers during a user interface development phase. In general, it supports the development of user interfaces in two different ways: First, researcher can use it to study user interfaces and its components for wearable computers. While doing so, researchers do not have to deal with underlying software infrastructures like, e.g., I/O device, rendering management, event queuing, or accessing context information. Instead, they can immediately concentrate on specific aspects to examine of the interface by using the WUI-Toolkit for rapid prototyping. Second, the toolkit provides application developers with an infrastructure to develop user interfaces needed for their wearable application by using a model-driven approach. Note, that for the scope of this paper we will concentrate more on latter case.

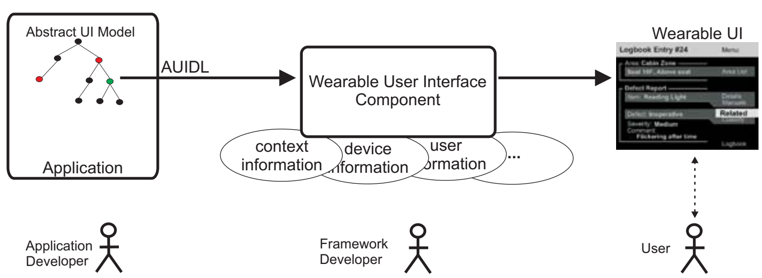

Fig. 3. Actors Using the WUI-Toolkit

Independent of its particular use, the WUI-Toolkit can be seamlessly integrated as a tool into any software development process (see previous section). Figure 3 shows the separation of concerns approach taken by the toolkit. For this approach, three different actors are involved:

Application developers specify the abstract model of the required WUI without paying attention to specific I/O devices or layouts. Instead, they focus on describing the interface by defining the needed input from the user and the associated output the user needs for a particular situation. If required, context based behavior that is not included in the toolkit's default setup can be implemented by using third party software, like e.g. the one described in [12]. For the application described in this paper, application developers integrated the possibility to scan RFIDs, e.g., for identification of the user.

In the center of the process there are the *framework developers* that provide the basic reusable functionality of the toolkit. This includes e.g. basic interface components, its layout or multi-modal interaction support.

Finally, the output generated by the toolkit is the concrete rendering of the abstract model specified by the application developer. It can be used by a *user* to control the wearable application.

3.1 Steps to be done by the Application Developer

The basic two steps to implement a WUI with the toolkit, from the perspective of an application developer, are:

1. Specification of an abstract model of the UI that is independent of any concrete I/O devices, interaction styles, or modalities to be used.
2. Configuration of the toolkits underlying software infrastructure including the configuration of the core framework by describing the capabilities of the wearable system to be used.

Note, that the configuration of the toolkit in the last step is not mandatory. It is only required if the default configuration of the toolkit does not meet the specific requirements of the needed UI. A reason could be a very specific layout or visual appearance required by the resulting interface that it not yet supported by default in the toolkit. For the seat removal application described here there was no need for a specific layout and thus application developers used to default configuration of the toolkit.

3.2 Abstract Model Specification

Because the abstract UI model (AUI) is independent of any I/O device or interaction style, it does not contain objects like buttons or windows, which are common to WIMP desktop applications. This is, because those WIMP interfaces are often not suitable in wearable applications [13]. Hence, the elements to represent information and interface elements with the AUI are more reduced than in many other UI toolkits. The application developer has to specify the envisioned interface in a dialogue or task based style, i.e. thinking in hierarchy of steps in sequential order. There are currently six elements available to specify such dialogues:

- **Information**:
  An *information* represents an atomic piece of information presented to the user. In most cases it contains text data, but may also contain multimedia content such as image, sound, video, or a combination of alternate contents. With the latter mechanism, the toolkit can offer different representations of the same piece of information that is most appropriate in a particular situation.

- **Group**:
  groups are used to structure information and contain other abstract elements (also other *groups*). For example, each user dialogue might be on the top level a group. The rendering of such groups depends again on the current output medium selected.
- **Selection List**:
  A *selection list* is composed of a set of *information* items and a set of *explicit triggers* which can be applied to *information* items in the list.
- **Explicit Trigger**:
  An *explicit trigger* represents an action by the user and requires an explicit interaction with the user. Typically, such actions change data, e.g., a "save file" would be specified by using an *explicit trigger*.
- **Explicit Text Input**:
  The *text input* element is used to input free text by the user. This would be the case, e.g., if a user has to enter a report.
- **Implicit Trigger**:
  An *implicit trigger* represents an implicit interaction that is either only indirectly triggered by the user or independent of the user. Typically, implicit interaction [14] is used to implement context-awareness of an application and basically allows to trigger action without explicitly requiring a user to do this. For instance, a recognized user activity of a context recognition system may trigger automatically an action.

Although the WUI-Toolkit is still under development, the expressiveness of the abstract model has already been shown as appropriate to model complex applications in different application domains (see e.g. [15, 6]).

Note, that with this approach application developers can successfully develop user interfaces for wearable computers even if they are not aware of the particular challenges of interaction and interface design in wearable computing.

4 Building the Application

By following the approach "start with the simple and go to the complex" the application developer took the opportunity to use the WUI-Toolkit not only to implement the final application but also to design the first mock-up. The concept of the WUI-Toolkit allows starting developing an application for wearables without taking care of any design or hardware issues. Only structural issues (AUI) have to be taken into account in the beginning of the development. In iterative steps the application gets more and more concrete by adding content, refining the structure and taking other constraints (e.g. hardware availability) into account.

The development team took the decision to work on a cabin scenario that is one of the subtasks of the "Removal and Installation Scenario" and deals with the removal and installation of a passenger seat in the cabin. This task offers on the one hand a potential of a benefit by introducing wearables and on the other hand the possibility of the implementation of innovative wearable concepts. The following steps give an overview of the task that has to be performed to remove a cabin seat:

– *Open scheduled procedures (Jobcard)*

1. Open circuit breaker to disable electrical entertainment system
2. Walk to seat
3. Remove seat-track covers
4. Mark position of seat
5. Disconnect electrical connector under the seat
6. Loose seat stops at the back of the seat
7. Loose screws at the front of the seats
8. Push seat backwards with hammer

– *Job close up*

After having designed the AUI first constraints were taken into account, such as content constraints, hardware constraints, and usability constraints. Content constraints were mainly caused by the Aircraft Maintenance Manual (AMM) that is only available in paper format for most of the aircrafts. The usability constraints strongly correlated with the content constraints in a way that usability requires appropriate representation of content in order to enable usability. For this reason, usability tests were performed previously by addressing the operators' needs. The results of the usability tests and the content constraints caused a creation of customized content in the form of different media (audio, video and images) that induced a rework of the WUI-Toolkit addressing the framework developer for integration of services that can deal with different types of media. In a very similar way the application developer dealt with the hardware constraints. Taking into account which I/O devices the final application will support should not be the focus of an application developer. Again, the application developer does not care about the I/O hardware devices, because the WUI-Toolkit handles the device issues like navigation devices and, e.g., HMDs (Head Mounted Display). While getting more and more concrete, other constraints arose that were solved either by the application developer or by the framework developer depending on the type of constraint.

Functionality of the application: As mentioned in section 4 the maintenance process consists mainly of eight logical steps (enclosed by the overview of the scheduled jobs and by the close up of the task) in the original maintenance procedure. The procedure was more and more enriched with the increasing number of mock-ups by taking the existing procedure and adding functionalities that arose from the requirements of the end user. Again, the overall goal is to support the user for the daily work by providing suitable functionalities within a wearable application. It should not provide only a wearable maintenance manual, moreover it should be possible for the user to focus more on the real work task than on handling the system itself.

For the implementation of the system, a OQO wearable computer was chosen to meet performance requirements. On this device a Windows XP Professional

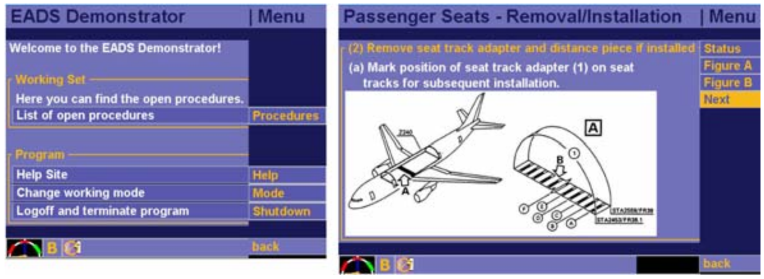

Fig. 4. Start Screen with scheduled jobs and Image/Drawings Output modality

was running as OS for the application. As physical input devices a USB microphone (speech navigation) and a dataglove (over Bluetooth) were connected to the system. The dataglove allows navigation through the navigation bar of the application as well as a identification of RFID chips that can be easily attached to items or persons. A MicroOptical SV-6 head-mounted display was used to present the visual interface.

The functionalities implemented can be classified as *workflow and context aware* or *UI and representation* oriented. The *workflow and context aware* oriented functionalities are indirectly targeting the UI through simplifying check tasks in the maintenance procedure, e.g., a system log on or the recognition (check) of an item. With the introduction of this type of functionality complex navigation becomes in some cases obsolete. It becomes clear by regarding the first step of the procedure *Open scheduled procedures (Jobcard)*: Computer based systems always require some kind of identification of the user. This happens because of the personalization of the system and primarily of security reasons. Performing the same task on a desktop system a user would simply type in the user name and the password. A maintenance operator equipped with a wearable system has only limited possibilities for entering confidental user information without a common keyboard. The implemented RFID-Reader functionality offers two ways of user support: (1) with the possibility of an automatic login step the user can simply log into the system and get the scheduled jobs for his shift (figure 4) by passing the dataglove over a personal RFID chip built in the operator's jacket. The RFID-Reader is also used (2) to identify items (like circuit breaker panels or seats) which have an inherent danger of confusion due to slightly different circuit breaker panels in aircrafts. The same counts for the seats. They look also very similar but can differ depending on the row they are located. The RFID functionality was also implemented for the seat identification.

The *UI and Representation* oriented functionalities are mainly implemented by the two input modalities (dataglove and the speech navigation) working together with the WUI representation of the content. The working conditions of maintenance operators in aircrafts vary strongly and are in some cases very inconvenient. A switching between the different input modalities becomes evident to give the user the freedom to have a hands free operation as required (with speech navigation). Over a single button on the dataglove the input modality can be easily switched over to the other. The user gets feedback of the current input mode via an icon on the WUI-application.

Usually, operators do not need the detailed maintenance manual due to their experience. If they are forced to follow every tiny step of the manual the application would have vast acceptance problems. As a consequence there is the need to adapt the level of detail in the representation. A function for switching between expertise profiles allows maintenance operators with different qualification levels to switch between top level descriptions and in depth descriptions of the task even while running the application.

5 Evaluation

The analysis and the evaluation of the results is divided into two parts: (1) The evaluation consists of the data interpretation of the performed usability studies consisting of the aspects mentioned in section 2. (2) The evaluation of the application is still an ongoing process and is not finished yet.

The usability outcomes showed for the different input devices an expected fast interaction with the use of the hand mouse. The interaction with the dataglove showed as well good results in such environments. The subjects felt quite comfortable with the dataglove, but gave also good hints for the improvement of the usability. They were disappointed about the difficult use of the speech navigation and its behavior especially with background noise.

Output modalities were rated best for the (still-) image output. The usability test results of the performance measurements prove this result, although the video output performed also well.

The evaluation of the whole application is still ongoing and was pretested with one subject so far (Figure 5). This first pretest gave information about the wearable application and integrated components that will be used for preparing an in depth usability study. It revealed information about the structuring of the content and the use of the dataglove. These outcomes will be taken into account for the upcoming tests at the workers' premises in order to gain valuable results for the application itself and the whole development process as well.

6 Conclusion

First results show on the one hand a successful result, in form of an application. Preliminary results suggest also good work on transferring requirements to application including the usefulness of the provided default layout and interaction styles by the toolkit. On the other hand the development process itself performed very well. The WUI-Toolkit approach speed up the development process of wearable application, although it is still under development. The basic idea to first formalize the programming of the UI with a structured abstract model to

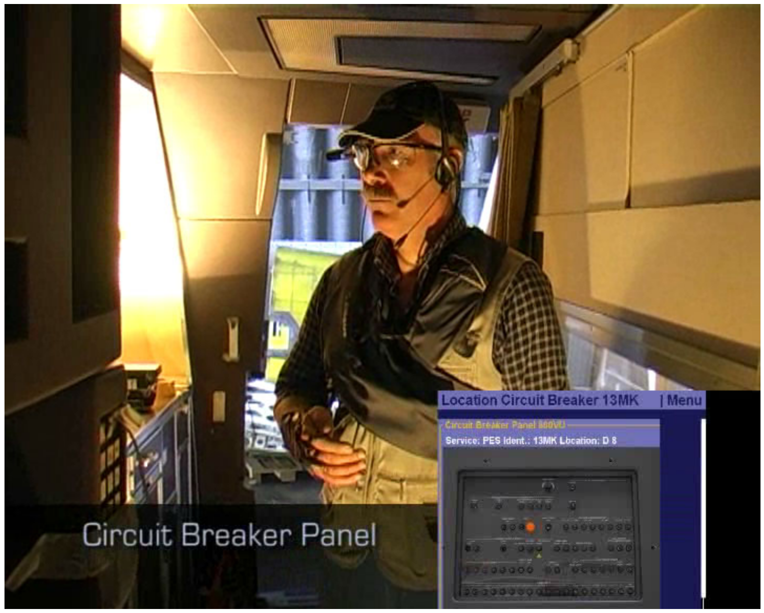

Fig. 5. Application Test and Evaluation

more easily split the content and the structure, the WUI-Toolkit can be introduced in an early stage of the development process and can be used continuously through the whole development process. Even though the development team had to cover different aspects of application development, e.g., requirements elicitation, content creation, classical application development and the integration of new services into the WUI-Toolkit the concurrent character of the WUI-Toolkit allowed a fast and in great parts parallel work of all team members.

6.1 Future Work

The very next steps are a deeper analysis of the application through an extended user study together with maintenance operators. This results will obviously affect first the interaction styles of the WUI-Toolkit as well as the services integrated in the WUI-Toolkit that provides these functionalities. To improve not only the UI of the application, but also to shift the toolkit itself to another level further investigation has to be performed focusing more on the application developer. The WUI-Toolkit will be introduced in other scenarios than the maintenance sector to gain experiences and feedback from a number of application developers. Optimizing the usability of the toolkit itself will accelerate the wearable application development once more.

References

1. Starner, T.: The challenges of wearable computing: Part 2. IEEE Micro **21** (2001) 54–67
2. Baber, C.: Wearable computers: A human factors review. International Journal on Human Computer Interaction **13** (2001) 123–145
3. Obrenovic, Z., Starcevic, D.: Adapting the unified software development process for user interface development. The international journal on computer science and information systems **3** (2006) 33–52
4. Bass, L., Kasabach, C., Martin, R., Siewiorek, D., Smailagic, A., Stivoric, J.: The design of a wearable computer. In: CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM Press (1997) 139–146
5. Boronowsky, M., Nicolai, T., Schlieder, C., Schmidt, A.: Winspect: A case study for wearable computing-supported inspection tasks. In: Fifth International Symposium on Wearable Computers (ISWC '01). Volume 5., IEEE Computer Society, IEEE Computer Society (2001) 163–164
6. Witt, H., Nicolai, T., Kenn, H.: Designing a wearable user interface for hands-free interaction in maintenance applications. In: PerCom'06: Proceedings of the 4th annual IEEE International Conference on Pervasive Computing and Communications, Pisa, Italy, IEEE Computer Society (2006)
7. Drugge, M., Hallberg, J., Parnes, P., Synnes, K.: Wearable systems in nursing home care: Prototyping experience. IEEE Pervasive Computing **5** (2006) 86
8. Haniff, D.J., Baber, C.: Wearable computers for the fire service and police force: Technological and human factors. In: ISWC '99: Proceedings of the 3rd IEEE International Symposium on Wearable Computers, Washington, DC, USA, IEEE Computer Society (1999) 185
9. Bo, G., Lorenzon, A., Chevassus, N., Blondel, V.: Wearable computing and mobile workers: The aeronautic maintenance showcase in the wearit@work project. In: Proceedings of the 3rd International Forum on Applied Wearable Computing (IFAWC), Bremen, Germany (2006)
10. : The wearit@work project: Empowering the mobile worker with wearable computing. IST Project (No. IP 004216-2004) (2004)
11. Witt, H.: A toolkit for context-aware wearable user interface development for wearable computers. In: ISWC'05: The 9th International Symposium on Wearable Computing. Doctoral Colloqium., IEEE (2005)
12. Sbodio, M.L., Thronicke, W.: Context processing within an open, componentoriented, software framework. In: Proceedings of the 3rd International Forum on Applied Wearable Computing (IFAWC), Bremen, Germany (2006)
13. Starner, T.: Attention, memory, and wearable interfaces. IEEE Pervasive Computing **1** (2002) 88–91
14. Schmidt, A.: Implicit human computer interaction through context. Personal and Ubiquitous Computing **4** (2000)
15. Nicolai, T., Sindt, T., Kenn, H., Reimerders, J., Witt:, H.: Wearable computing for aircraft maintenance: Simplifying the user interface. In: Proceedings of the 3rd International Forum on Applied Wearable Computing (IFAWC), Bremen, Germany (2006)