Artificial Intelligence in Wearable Computing – Perspectives and Challenges

Holger Kenn, Tom Nicolai, Hendrik Witt, Stéphane Beauregard,
Michael Lawo, Michael Boronowsky, Otthein Herzog

Research in wearable computing has pursued the goal to build a ubiquitous wearable assistant for more than a decade. Although significant progress was made, key challenges remain still unsolved. In this paper, we suggest the application of AI techniques in wearable computing that may build a path towards successful wearable systems and their applications in four areas; user interfaces, knowledge management, context awareness and cooperation through agent-based computing. By applying AI techniques in design and implementation, wearable computing applications can be made adaptable, user- and use-centric at the same time. In turn, these will then result in better end user acceptance.

1 Introduction

The research area of wearable computing has been active for a couple of years now, but only few examples of successful large-scale implementations of wearable computing applications such as [SFH$^+$98] are known. There are probably many causes for this. Often, the benefit of wearable computing was overestimated, while the necessary effort to implement a wearable computing solution was underestimated. In this paper, we argue that the application of AI techniques may show a path to solve both problems in the implementation of successful wearable computing applications.

The defining character of a wearable computing system is the fact, that working with the computer is not the primary task of the user. Although it is possible to run standard application software on wearables, it is often not desirable. Usability of standard applications is limited by the existing input and output devices, the unfitness of the WIMP (Windows, Icons, Menus, Pointer) user interfaces and the fact that such interfaces require the user's full attention. Moreover, acceptance of such computers is low when they capture the attention of the user, directing it away from real-world tasks or events or human dialog partners. In our experience, in a successful wearable computing application, the computer takes the secondary task, supporting a primary task that takes place in the real physical world surrounding the user. To support the user efficiently, the wearable has to keep track of the user's primary task, providing information when needed, limiting explicit human computer interaction and becoming truly wearable in the sense of clothing that provides its protective function without requiring neither attention nor interaction.

As commercial products such as the one described in [SFH$^+$98] show, it is possible to build and optimize a wearable computer system for a single task and situation. However, since users often have to perform multiple tasks in various situations, a system has to be designed for and tested in all these situations performing all these tasks. Even for a simple set of tasks and situations, it becomes tedious and expensive to perform this optimization. The application of AI techniques provides an alternative. By the specification of tasks on an abstract level, an application can be adapted automatically to fit the user's current task and environment.

This approach is suitable for ambient intelligence in general, but in wearable computing, the goal is to implement these functions in a device worn by the user instead of devices in the environment. The two approaches complement each other. Wearable computing is especially suitable for spaces that cannot be instrumented easily, such as large-scale industrial environments, and for applications in unknown environments, such as firefighting and urban search-and-rescue. The wearable nature of the devices pose well-known limitations [Sta01] that have to be taken into account.

We will illustrate our approach in four areas of wearable computing research: adapting user interfaces, knowledge management for wearable assistants, context awareness and cooperation through multi-agent systems.

2 User Interfaces for Wearables

Today, WIMP user interfaces are ubiquitous. They dominate desktop computers and even mobile devices. Although the use of the WIMP metaphor on mobile devices is often a burden, it is usually reused. Consequently, it lacks usability when the user focuses on a primary task [Sta02]. E.g., to control a mouse pointer, constant visual attention to the interface is required. However, it is impossible to spend a lot of attention on the UI, when a primary task has to be performed as well. Therefore, wearables call for new user interfaces that regard its unique characteristics of being involved in a mobile, dual-task situation while having the possibility to make use of available context information, the user's activity, and environmental conditions.

To successfully build wearable user interfaces (WUIs) there are a number of AI techniques available that can be applied. In particular, research in the fields of *intelligent user interfaces* (IUI) and *multi-modal interfaces* have yielded already many useful techniques and approaches.

2.1 Semantic Modeling Languages

Due to the highly dynamic environment in which wearable computing takes place, environmental conditions and user activities often change over the time an application is used. E.g., changing light conditions impact the readability of information on a head-mounted display (HMD), similar to reading text on a notebook when being outdoors on a sunny day. A similar situation can occur when speech interaction is used in noisy environments. If background noise in a certain situation is too high, recognition errors will likely increase. Then, a dynamic change of the interaction method used would allow for warranting usability. Hence, intelligent UIs that adapt automatically by taking available context information into account, are deemed to be valuable in terms of usability.

An abstract description of the UI, independent of I/O devices and interaction paradigms, is a fundamental approach to achieve an "intelligent" alteration of the interface. By inferring its semantic properties under a given situation, an optimized interface can be generated (see [SJ04] for an overview). Instead of programming one specific WUI in the usual "hard-wired" way, an abstract and thus flexible description of the interface, that describes the semantics of an interface instead of specifying its concrete representation and interaction style, is preferable. For this purpose, different model-based approaches have been proposed (see e.g. [TCC04, FDM04, Pat99]). They usually rely on special modeling languages that provide the interface description. Although these are supposed to be independent of its representation they often suffer from being designed with having the WIMP metaphor in mind. Languages and systems usually implement a "write once run anywhere" approach that only considers device adaptation. To be applicable for wearable computing, extensions are needed that allow interaction beyond the WIMP metaphor and easy integration of context-awareness based on sensory information of the local system or distributed sources.

2.2 Constraint Satisfaction

When considering UI adaptation beyond output devices, rendering and layout of interface components becomes challenging. In a wearable computing environment many constraints are given that influence a rendering procedure. These likely exceed those of mobile or desktop environments which typically focus on static aspects such as display resolution or size. Constraint satisfaction techniques are deemed to support these kind of problems.

E.g., SUPPLE [GW04] uses constraint satisfaction techniques and treats the layout and generation of UIs based on an abstract description as an optimization problem. Although, it has been argued that automatic UI generation has not yielded satisfactory results for complex UIs yet [SJ04], WUIs are usually rather simple in structure to be easily controllable even in a dual-task situation. Hence, constraint satisfaction is a promising technique to implement automatic adaptation. Even though the technique itself is promising, constraints are needed that describe the coherency of interface components, layouts, interaction styles, and I/O devices.

Today, application designers can rely on a large amount of UI widget libraries, known and tested UI designs, and usability guidelines. Unfortunately, these are usually not applicable to wearable UI design [WNK06]. So far, design knowledge for WUIs only exists in the form of system descriptions of more or less successful implementations or comparative studies for single UI elements (e.g. [LPS03, BNSS01]). Unlike traditional applications, wearable applications are currently still implemented from scratch, because there are no common tools or frameworks available, yet.

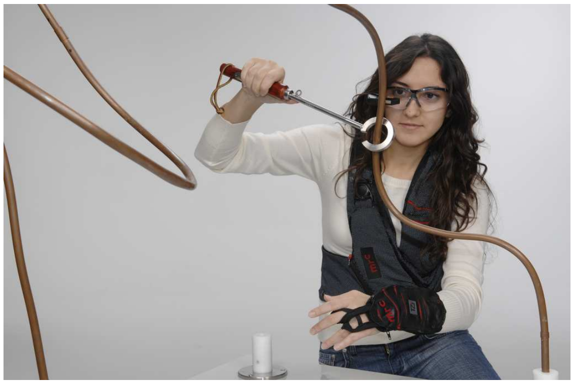

Figure 1: The HotWire primary manual task simulator.

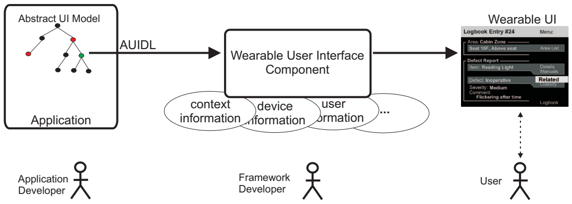

Figure 2: Interface generation process of the WUI-Toolkit.

When considering desktop applications, most constraints can be derived from research in human-computer interaction (HCI). For WUIs, however, this approach has pitfalls. Research in wearable computing has to focus on the establishment of a common understanding of how interaction has to be designed to be applicable in dual-task situations where the computer is only secondary. E.g., figure 1 shows the so-called HotWire apparatus [WD06]. It provides a possibility to simulate a primary real-world task while retaining the properties of wearable computing in a laboratory environment. A user has to pass a ring attached to a special tool over an intricately bent metallic wire from one end to the other without touching the wire. The apparatus provides a method that facilitates the evaluation of interface components for wearable computers. Outcomes of such kinds of evaluations that directly focus on wearable computing properties, are likely to be more valid in interface design than a direct adaptation of desktop knowledge. In turn, these finding can then be used to formulate general constraints about a certain aspect of the WUI needed during an interface generation or adaptation process.

2.3 Reusing Available Knowledge

To further support the process of establishing basic properties of a WUI, acquired knowledge has to be stored in a way that makes it reusable in other applications similar to style guides or widget libraries for desktop environments.

The Wearable User Interface Toolkit (WUI-Toolkit) [Wit05] provides a kind of knowledge base that application developers can use even with a limited understanding of wearable computing. It offers a tool set that supports the development of

WUIs with reusable components to reduce implementation efforts. Therefore, the WUI-Toolkit utilizes a model-driven approach, where the envisioned UI is specified in an abstract, i.e. device- and context-independent, way (see figure 2). During interpretation, the toolkit tries to find a corresponding representation of the UI by solving a constraint satisfaction problem. For this, environmental context and user context as well as design constraints obtained from experiments, e.g., with the HotWire, are taken into account. Finally, a concrete WUI is generated. The WUI-Toolkit is currently developed as a central part in the WearIT@work project [wea04] in which the establishment of an European Wearable Computing Framework (EWCF) is envisioned.

3 Knowledge Management in Wearable Computing

As described in the last section, the development of a WUI is limited by the available cognitive resources of the user and the restricted in- and output modalities of the wearable computer. Complementary to the UI techniques of the last section, the user experience can be improved with a knowledge management system that provides contextual access to information and services.

3.1 The Smart Wearable

The development of wearable computing systems is often guided by the metaphor of the smart assistant. A human assistant is expected to closely watch his master and understand what he is doing. Thus, he is able to look up relevant information and can even anticipate which tools and information might be needed in the next step. The basis of a wearable smart assistant is the ability to detect the context of its user. Queries like “Show the part list of this area of the aircraft cabin” or “Show defect entries that are related to the current” can only be resolved with the appropriate knowledge about the context of the user—in this case the location of the user inside an aircraft or the defect he is currently working on, respectively. Compared to desktop programs, queries involving the current position and activity of a user are a new challenge.

On top of the knowledge about context, proactive mechanisms can be implemented. The ability to act pro-actively is one of the defining criteria of a wearable computer and regarded as a key to its successful application [Sta01]. Without pro-activity, the user is forced to switch back and forth between primary task and interaction with the computer, whenever he needs additional information. If there are strict working procedures and sequences, pro-activity can be incorporated to realize an automatic push of contextual information. When the wearable detects that a working step is complete, it can automatically display instructions for the next step, e.g. the force required to fix a part or security advice that applies in the current situation.

3.2 World Model in Context

To provide appropriate information depending on the current context, the wearable can relate the context to a model of the working environment. The location of the user is an obvious piece of context information for working environments where the user is required to move between work places, parts or machines. Different localization techniques can be deployed. Depending on the required precision, area of coverage and price there are several technologies available, including ultrasonic or RFID. A corresponding mapping is required to relate a location with resources like troubleshooting manuals or task lists. A spatial model can provide further information based on the knowledge about the spatial relation of places.

The current activity of a worker is also important. Sensors close to the hands, on other parts of the body or in the environment may be used to recognize his current movements. Signal processing and machine learning is used to detect which tool a worker uses. Together with knowledge about the possible working processes, the wearable can then infer which task its user probably works on. Additionally, sensors in the environment might monitor components and provide input to the model, components with self-diagnosis functions can directly be connected to the model. Apart from reasoning about useful information for the user, the status of the model can also be visualized directly. With Augmented Reality displays, hidden and covered components can be displayed selectively depending on the current point of view [Miz01].

3.3 Aircraft Maintenance Use Case

The use of wearable computing in professional environments is an applied research method that has gained attention in the recent years. The application of wearable computing can lead to increased labor efficiency and thus is a tool to reduce cost. As an application domain for professional wearable computing, we chose the maintenance of large passenger aircrafts. The airline industry is characterized by significant cost pressure that has increased over the recent years with the increase of fuel prices. A key figure in the economically efficient operation of an aircraft is the ratio of actual in-flight time versus ground time. In a project with a major European aircraft manufacturer, we designed and implemented a wearable computing system to reduce the time required for maintenance as a test-bed for the evaluation of a complete wearable computing system [NSW$^+$06].

The wearable is supported by a knowledge management system based on the KnowWork system [TALS01]. There are two different kinds of models within the aircraft. First, there is a classification of parts, which is called the product model. The second model contains the particular parts and their relationships among each other. This is called the product structure. These two models are combined to a complex ontology describing the whole aircraft. Each part of the aircraft is an instance of a concept, which is defined within the classification. Each concept is set into relationship with one or more documents. Each part may be associated with a RFID tag to implement location-sensitive queries. The ontology is stored on a maintenance servers on board of aircrafts with variations representing the specific aircraft type and layout. When the maintenance crew enters the aircraft, their wearables connect wirelessly to the server and access the aircraft’s specific data.

We found that large and unstructured lists of defect reports tend to be confusing on the HMD of a wearable computer. With the structural knowledge about the aircraft, the system is able to cluster different defect reports with a common cause. E.g., the aircraft diagnostic system reports a malfunctioning transformer and a flight attendant reports about defect lighting. If there is a close structural proximity of the two parts, it is important that the technician is informed about their connection. The cluster mechanism generates structure automatically and defect reports can be presented to the user as a cohesive group of reports. Thus, the knowledge-based approach saves time and improves the usability of the system.

Another problem in maintenance systems is that the experience knowledge of colleagues is usually not accessible. Because there are certain problems that occur repeatedly, the maintenance personnel can save time by having direct access to previous repair reports. Beyond standard troubleshooting procedures, these reports are a guidance to the specific problems of an aircraft. Additionally, it is efficient to copy and slightly adapt an existing repair report to document the repair procedure. We complemented the wearable maintenance system with a similarity search engine to quickly locate relevant reports in an experience knowledge base. Whenever the user reads a defect report, the system locates similar past defect reports and displays the related repair report on request. Instead of standard procedures, the user has access to specific reports and saves time when writing the new repair report.

The contextual access to information in wearable computing systems is a critical part with the opportunity to dramatically improve maintenance and repair procedures as demonstrated in the aircraft maintenance domain. Sensors on the body of the user and in the working environment relate actions of the user to working processes and localize data in space.

4 Context in Wearables

In order for wearables to provide a pro-active capability and get the right information to the user at the right time (e.g. a specific page of an assembly manual), the wearable must have context-awareness. The location of the user, in absolute terms, but more importantly relative to key landmarks, is an essential feature of context-awareness. Additional to location, a wearable must also be sensitive to general environmental conditions such as ambient noise, lighting, temperature, humidity and so on. The user's physiological state (heart rate, stress level), body attitude and current task (e.g. drilling, hammering) are also very important as they may have a direct bearing on the user's interruptibility.

4.1 Training Data vs. *a priori* Knowledge

The tools and techniques available from the domain of statistical machine learning (ML) [Bis95, DHS00] are in principle very useful for recognizing context from multi-sensor data. However, some caution must be exercised in using ML techniques. First of all, wearables are very often power-constrained devices so running sophisticated probabilistic algorithms is typically not an option. Second, ML techniques can require substantial amounts of training data. If this data must be acquired with the user's participation, not only is this inconvenient, it may take some time before typical clustering / classification algorithms give reliable results. To get around both these problems, one can leverage varying degrees of *a priori* knowledge about the system under consideration [NGP98].

4.2 Example: Activity Recognition

At one end of the spectrum, if the analyst does not want to do any system modeling, the *agnostic* application of ML can work well given sufficient training data. This approach can be applied to activity recognition. Using a few simple sensors, such as an EKG sensor and accelerometers in a chest strap, and minimal preprocessing, it is relatively easy to classify physical activities, such as walking, running, sitting and standing, with high accuracy using a general-purpose ML algorithm [LCK$^+$05]. The resulting classifiers are however brittle, in the sense that there are no guarantees that they will behave correctly with unusual input not seen in the training data.

At the other extreme, if the analyst has a deep understanding of the system under investigation, complex mathematical models can often be built. In this case, the system behavior can be specified by a few parameters having real physical meanings. The optimal values of the parameters can usually be determined with very little training data from carefully-chosen, calibrated sensors. E.g., with multiple wearable motion measurement units, joint constraints and appropriate kinematic models, poses, gestures and locomotion can be identified with fine granularity. The ML part of this approach could be the estimation of segment lengths and sensor positions using general-purpose parameter optimization routines. The outcome of this modeling effort is a very flexible system, enabling extensible activity descriptions. In between these two extremes is the more frequent situation where extensive prior knowledge of the system is available but might not be implemented easily. In this case, complex systems can often be reduced to much simpler models with lumped parameters. For example, the equations describing the intricate biochemical pathways in play during physical exercise could be reduced to a much simpler model with a small number of parameters [Rid91]. The optimal values of these parameters could be determined from recorded test data. This type of model might be sufficiently accurate for estimating when fatigue thresholds are crossed during intense physical activity (e.g. fireman search and rescue mission). Another example is step length estimation in pedestrian dead reckoning (PDR) systems [Lad02]. In principle, one could estimate step lengths using a complicated biomechanical model of the lower limbs, motion sensors on several segments and contact switches on the shoes. It turns out however, that there is a very strong empirical and theoretical relationship between step frequency and walking speed [Kuo01, GMK04]. Consequently, very few sensors or training data are actually required to get accurate step length estimates [LKST02]. The step frequency can be measured with sensors mounted at various locations on the body [Bea06], giving the implementor great flexibility.

This section shows, that there is a large design space for context recognition systems in wearables and no single approach will fit all applications. After all, an activity recognition subsystem for consumer smart phones will look very different from one built for firemen. The choice between an "agnostic" approach or high-fidelity engineered modeling (or somewhere in between) ultimately depends on end-application requirements and constraints, such as power, size, weight, cost, accuracy and reliability.

5 Multi-Agent Systems Coordinating Wearable Agents

Large scale urban disaster situations such as earthquakes, floods or terrorist attacks pose an important threat to modern urban civilization centers. Examples such as the Kobe earthquake, the pacific tsunami, the hurricane Katrina or the tragic events of September 11th 2001 demonstrate that although preventive measures and disaster response plans were in place, these events pushed existing countermeasures over their limits, resulting in loss of human lives, chaotic situations and long term adversary effects on the affected regions.

After the Kobe earthquake, the Japanese government decided to promote researchers to work on problems related to large-scale urban disasters. One of the outcomes of this initiative are the RoboCup Rescue competitions. Using the same successful method of competition-based benchmarks that the Robot Soccer competition was applying, the RoboCup Federation added two new competitions to the RoboCup, RoboCup Rescue Robot League and RoboCup Rescue Simulation League.

The Rescue simulation league aims at simulating large-scale disasters and exploring new ways for the autonomous coordination of rescue teams [KTN$^+$99]. In the Rescue Simulation league, the goal of a team participating in the competition is to provide a software system that reacts to a simulated disaster situation by coordinating a group of simulated agents such as police, ambulance and fire brigade agents. This goal lead to challenges like the coordination of heterogeneous teams with more than 30 agents, the exploration of a large-scale environment in order to localize victims, as well as the scheduling of time-critical rescue missions. Each of these agents only has a limited amount of communication bandwidth they can use to coordinate with each other, so the problem cannot be addressed by a central coordination entity, but has to be solved by a true multi-agent system. Moreover, the simulated environment is highly dynamic and only partially observable by a single agent. Agents have to plan and decide their actions asynchronously in real-time. Core problems are *path planning*, *coordinated fire fighting*, and *coordinated search and rescue* of victims.

The solutions presented in this paper are based on the Open-Source agent software[1], which was developed by the *ResQ Freiburg 2004* team [KBB$^+$05], the winner of RoboCup 2004. Applying this simulation system in the real world currently lacks the interface to the real world information, i.e., the simulation system relies on carefully designed special-purpose map data and observations generated by the simulation itself, e.g., agent motion is computed by a traffic simulator and cannot be observed from real-world motion of responders. However, by using wearable computing technology, we can provide such observations to the simulation system. This has three uses. First, it can be used to record real-world data from real-world intervention scenarios and by this, assess the validity of the simulation. Second, it can be used to observe the reaction of team multi-agent systems to real-world data. Third, it is a step towards using both the simulation system and the team multi-agent systems to support incident commanders and responders in training and real interventions by providing autonomous decision support and faster-than-realtime simulation for strategy decisions.

We have implemented a number of wearable computing devices, acquiring disaster relevant data, such as locations of victims and blockades, and show the data integration into the *RoboCup Rescue Simulation* platform, which is a benchmark for MAS within the RoboCup competitions. We have shown exemplarily, how the data can consistently be integrated and how rescue missions can be optimized by solutions developed on the RoboCup Rescue simulation platform [KBK06]. These preliminary results indicate, that nowadays wearable computing technology combined with MAS technology can serve as a powerful tool for Urban Search and Rescue (USAR).

Our approach consists of wearable devices that can localize themselves with sufficient accuracy and can cope with an unreliable, bandwidth-limited, IP-based communication channel. Communication is currently implemented through commercially available mobile phone networks, but neither availability, bandwidth, nor latency are critical for our application. The search for victims of many rescue teams can only be coordinated efficiently if the rescue teams share information on exploration. We assume that rescue teams report when they have finished to explore a building and when they have found a victim, by transmitting the according message to the command center. The command center utilizes this information for distributing rescue teams efficiently among unexplored and reachable locations. In the future, the system will be tested in real-world training events and observations and user experiences will be used to further investigate the suitedness of our approach for urban search and rescue.

6 Conclusion and Future Work

In this paper, we have shown examples for the successful application of AI techniques in wearable computing in the four areas of adapting user interfaces, knowledge management for wearable assistants, context awareness and cooperation through multi-agent systems. In these areas, the application of AI techniques provides a significant advantage for the user of a wearable system. Moreover, some of the application areas of wearable computing cannot be implemented without relying on these AI techniques. Therefore, we conclude that the application of AI techniques in wearable computing is a key factor for the success of wearable applications. In addition, wearable computing applications provide an excellent test field for AI techniques, especially in areas where the performance of a user is a direct result of the cooperation between man and machine, ranging from optimization such as adaptation of user interfaces and context recognition to complex situations such as wearable-mediated knowledge management and cooperation.

The application of AI techniques in wearable computing paves the way towards the ubiquitous use of AI in the day-to-day life of mobile computer users. However, there are a number of open research questions. As wearable computing tries to minimize the use of external infrastructure, an important question is how to implement complex AI techniques with limited energy, computation and bandwidth resources. Improvements in mobile hardware ease the situation, but most available mobile hardware and system software improves energy usage by limiting compu-

---

[1] ResQ Freiburg 2004 source code, available on: `http://gkiweb.informatik.uni-freiburg.de/~rescue/sim04/source/resq.tgz`, released September 2004.

tational resources, i.e., reducing CPU clock speeds or putting devices to sleep altogether, which contrasts the always-on requirement for context-awareness and activity monitoring.

Another open question is the response of users to devices that adapt to context and task. Will it improve user efficiency or add to the cognitive load of the user and lead to user confusion instead? These issues will have to be investigated in more detail in the future.

Acknowledgments

This work has been partially funded by IST project "WearIT@work: Empowering the mobile Worker with Wearable Computing" (IP 004216-2004). The authors wish to thank Thomas Sindt of TZI for his research contribution on knowledge management and Alexander Kleiner of Universität Freiburg for his cooperation on multi-agent systems research.

References

[Bea06] Stéphane Beauregard. A Helmet-Mounted Pedestrian Dead Reckoning System. In Otthein Herzog, Holger Kenn, Michael Lawo, Paul Lukowicz, and Gerhard Tröster, editors, *Proceedings of the 3rd International Forum on Applied Wearable Computing (IFAWC 2006)*, pages 79–89, Bremen, Germany, March 15-16 2006. VDE Verlag.

[Bis95] Christopher M. Bishop. *Neural networks for pattern recognition*. Oxford, 1995.

[BNSS01] Michael Boronowsky, Tom Nicolai, Christoph Schlieder, and Ansgar Schmidt. Winspect: A case study for wearable computing-supported inspection tasks. In *Fifth International Symposium on Wearable Computers (ISWC '01)*, volume 5, pages 163–164, Zürich, 8–9 October 2001. IEEE Computer Society, IEEE Computer Society.

[DHS00] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[FDM04] Peter Forbig, Anke Dittmar, and Andreas Müller. Adaptive task modelling: From formal methods to XML representations. In Ahmed Seffah and Homa Javahery, editors, *Multiple User Interfaces*, pages 171–192. John Wiley & Sons Inc., 2004.

[GMK04] S. A. Gard, S.C. Miff, and A. D. Kuo. Comparison of kinematic and kinetic methods for computing the vertical motion of the body center of mass during walking. *Human Motion Science*, (22):597–610, 2004.

[GW04] Krzysztof Gajos and Daniel S. Weld. Supple: automatically generating user interfaces. In *IUI '04: Proceedings of the 9th international conference on Intelligent user interface*, pages 93–100, New York, NY, USA, 2004. ACM Press.

[KBB+05] A. Kleiner, M. Brenner, T. Braeuer, C. Dornhege, M. Goebelbecker, M. Luber, J. Prediger, J. Stueckler, and B. Nebel. Successful search and rescue in simulated disaster areas. In *In Proc. of the International RoboCup Symposium '05*, 2005.

[KBK06] Alexander Kleiner, Nils Behrens, and Holger Kenn. Wearable computing meets multiagent systems: a real-world interface for the robocuprescue simulation platform. In Nicholas R. Jennings, Milind Tambe, Toru Ishida, and Sarvapali D. Ramchurn, editors, *Proceedings of the First International Workshop on Agent Technology for Disaster Management at AAMAS06*, 2006.

[KTN+99] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. RoboCup Rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE Conf. on Man, Systems, and Cybernetics(SMC-99)*, 1999.

[Kuo01] Arthur D. Kuo. A simple model of bipedal walking predicts the preferred speed-step length relationship. *Journal of Biomechanical Engineering*, 123:264 – 269, June 2001.

[Lad02] Quentin Ladetto. In Step with INS: Navigation for the Blind, Tracking Emergengy Crews. *GPS World*, pages 30–38, October 2002.

[LCK+05] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A hybrid discriminative-generative approach for modeling human activities. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI 2005)*, Edinburgh, Scotland, July 2005.

[LKST02] Helena Leppakoski, Jani Kappi, Jari Syrjarinne, and Jarmo Takala. Error analysis of step length estimation in pedestrian dead reckoning. In *ION GPS 2002*, pages 1136 – 1142, Portland, OR, September 24 -27 2002. The Institute of Navigation.

[LPS03] Kent Lyons, Nirmal J. Patel, and Thad Starner. KeyMenu: A keyboard based hierarchical menu. In *ISWC '03*, pages 240–241, White Plains, NY, October 2003. IEEE Computer Society.

[Miz01] D. Mizell. Boeing's wire bundle assembly project. In W. Barfield and T. Caudell, editors, *Fundamentals of wearable computing and augmented reality*, pages 447–467. Lawrence Earlbaum Associates, New Jersey, 2001.

[NGP98] P. Niyogi, F. Girosi, and T. Poggio. Incorporating prior information in machine learning by creating virtual examples. *IEEE Proceedings on Intelligent Signal Processing, 86(11):2196-2209*, 86(11), November 1998.

[NSW+06] Tom Nicolai, Thomas Sindt, Hendrik Witt, Jörn Reimerdes, and Holger Kenn. Wearable computing for aircraft maintenance: Simplifying the user interface. In Otthein Herzog, Holger Kenn, Michael Lawo, Paul Lukowicz, and Gerhard Tröster, editors, *IFAWC '06*. VDE Verlag, 2006.

[Pat99] Fabio Paternò. *Model-Based Design and Evaluation of Interactive Applications*. Springer Verlag, 1999.

[Rid91] V. Rideout. *Mathematical and Computer Modeling of Physiological Systems*. Prentice Hall, New Jersey, 1991.

[SFH+98] R. Stein, S. Ferrero, M. Hetfield, A. Quinn, and M. Krichever. Development of a commercially successful wearable data collection system. In *ISWC '98: Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, page 18, Washington, DC, USA, 1998. IEEE Computer Society.

[SJ04] A. Seffah and H. Javahery. *Multiple User Interfaces Cross-Platform Applications and Context-Aware Interfaces*. John Wiley & Sons, West Sussex, UK, 2004.

[Sta01] Thad Starner. The challenges of wearable computing: Part 1. *IEEE Micro*, 21(4):44–52, 2001.

[Sta02] Thad Starner. Attention, memory, and wearable interfaces. *IEEE Pervasive Computing*, 1(4):88–91, 2002.

[TALS01] H. K. Tönshoff, R. Apitz, A. D. Lattner, and C. Schlieder. Knowwork - an approach to co-ordinate knowledge within technical sales, design and process planning departments. In *7th International Conference on Concurrent Enterprising*, pages 231–239, 2001.

[TCC04] David Thevenin, Joelle Coutaz, and Gaelle Calvary. A reference framework for the development of plastic user interfaces. In Ahmed Seffah and Homa Javahery, editors, *Multiple User Interfaces: Cross-Plattform Applications and Context-Aware Interfaces.*, chapter 3, pages 29–51. John Wiley and Sons Ltd., West Sussex, UK, 2004.

[WD06] Hendrik Witt and Mikael Drugge. Hotwire: an apparatus for simulating primary tasks in wearable computing. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 1535–1540, New York, NY, USA, 2006. ACM Press.

[wea04] The wearit@work project: Empowering the mobile worker with wearable computing. IST Project (No. IP 004216-2004), 2004.

[Wit05] Hendrik Witt. 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.

[WNK06] Hendrik Witt, Tom Nicolai, and Holger Kenn. 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, 2006. IEEE Computer Society.

Contact

Dr. Holger Kenn
Technologie-Zentrum Informatik (TZI)
Universität Bremen
Am Fallturm 1
28759 Bremen
Tel.: +49 (0)421-2183035
Fax: +49 (0)421-2187196
Email: kenn@tzi.de

Bild

Since 2004, **Dr. Holger Kenn** is leading the scientific research of the mobile and wearable computing research group of Prof. Herzog at the TZI at Universität Bremen.

Bild

Hendrik Witt is PhD student at the TZI at Universität Bremen. His thesis is about the development and evaluation of user interfaces for wearable computers.

Bild

Tom Nicolai is PhD student at the TZI at Universität Bremen. His thesis is about the automatic recognition of social context.

Bild

Stéphane Beauregard is PhD student at the TZI at Universität Bremen. His thesis is in the context of "wearable" pedestrian navigation systems, positioning and machine learning for context recognition.

Bild

Prof. Dr. Michael Lawo is the Technical Manager of the IP 2004-004216 WearIT@Work. He received his PhD from Universität Essen in 1981 and became professor in structural optimization there in 1992. Since then, he held multiple top management positions in the computer industry until he joined TZI in 2004.

Bild

Dr. Michael Boronowsky is the managing director of the TZI since 2002. He is one of the founding members of the TZI WearLab and shaped the research agenda of the TZI and the WearIT@Work project. He received his PhD from Universität Bremen in 2001.

Bild

Prof. Dr. Otthein Herzog holds the position of a chaired professor in the Department of Mathematics and Computer Science at the University of Bremen, where he directs the TZI, the Collaborative Research Center "Autonomous Logistic Processes" (SFB 637), and the Mobile Research Center. Since he was appointed to his current position in 1993, Dr. Herzog built up the Artificial Intelligence Research Group. Before joining academia, he worked within IBM in industrial software development for 16 years.