IASO – An Activity-Based Computing Platform for Wearable Computing

Jakob Eyvind Bardram and Jonathan Bunde-Pedersen
Centre for Pervasive Healthcare
Department of Computer Science, University of Aarhus
Aabogade 34, 8200 Aarhus N, Denmark
{bardram,jbp}@daimi.au.dk

Abstract

Displaying and navigating complex information on wearable computers is very different from accessing the same information while sitting at your desk. Limitations in hardware, screen-size, input- and output devices are not met with changes in the software running the wearable system. We propose using the ABC-framework as a middleware layer providing collaboration, adaptation, activity sharing and context-awareness for wearable applications. Furthermore, enhancements in the user-interface and interaction techniques overcome some of the hardware limitations. The proposed system makes wearable computers usable by e.g. emergency workers, policemen and firefighters.

1 Introduction

Wearable computing is an emerging technology which enables the use of computers and access to digital material in working situation that are physically and conceptually far away from the office desk. Despite an intensifying research endavour, wearable computing still faces a number of challenges before wearable computing hardware and software melt naturally into everyday activities like police work, military warfare, maintenance work, and paramedic work. Some of these challenges involve basic hardware issues [9, 17]; other challenges involve more basic software architectures and operating systems for wearable computing [7]; and yet other challenges involve how users can use such computers for information retrieval, overview, data input and collaboration [4, 5, 16, 11]. In this paper we want to address the challenges in helping users navigate, overview, and interact with complex information on a small wearable screen; how to make a bridge between the user's current work activity and the working context; and how collaboration in wearable computing setup could be accomplished.

Activity-Based Computing (ABC)[1] have been proposed by researchers as a new computing paradigm for pervasive computing [6, 2, 18]. This paradigm helps users organize the computational support in logical bundles specific to a human activity and helps users to share such activities in a collaborative effort. Activity-Based Computing can be viewed as a pervasive computing platform – or operating system – with a underlying distribution middleware, a resource and context aware adaptation architecture, and a new user-interface metaphor. Furthermore, it contains a collaborative environment for synchronous and asynchronous collaboration.

In this paper we demonstrate how Activity-Based Computing can constitute a computational platform for wearable computing. This work is motivated partly by our current empirical work designing wearable computing support for emergency workers, like paramedics and ambulance personnel, and partly by a review of the research done within the field of wearable computing. Section 2 presents a future scenario for supporting an accident using wearable computers, and section 3 presents and discusses four design goals and principles for Activity-Based Wearable Computing. The paper then presents 'Iaso', which is an Activity-Based Wearable Computing platform with special emphasis on new types of user-interfaces, sharing of many displays, context-aware activity support, and support for collaboration[2]. Section 5 presents details on the implementation and the deployment of the Iaso platform, and section 6 concludes the paper by discussing to what extend the Iaso platform meets the design principles stated in section 3.

2 A Bus Accident Scenario

The 911 call center receives an alarm on a bus accident and immediately dispatches 5 ambulances to the scene of the accident. When the alarm goes off in the ambulance

[1] Also known as 'Task-centered computing'
[2] Iaso was a Goddess of Healing and the daughter of Asclepius.

Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)
1545-0678/05 $20.00 © 2005 **IEEE**

garage, the paramedic Dr. Strong and the other ambulance personnel grab their uniforms with a built-in wearable computer. While driving to the scene, they prepare for the accident by uploading maps of the scene and by loading medical data on the people involved in the accident. All this information is prepared for easy access and display later. When arriving to the scene Dr. Strong starts by walking around to all the injured people to create an overview of the accident. He attaches an RFID tag visible on all patients. This tag works as a temporary identification and he links this unique id with the patient's set of medical data which have already been prepared. The ambulances at the site create local WLAN coverage and has 3G connectivity back to the call center.

By using the speech interface, Dr. Strong is able to toggle between the different patients and to zoom in on specific information. While he is visiting the patients he (along with the other paramedics) are filling in the 'trauma forms', which indicate how, where, and how bad each patient is hurt. All this information is relayed to the ambulance server and on to the call center.

The work load for Dr. Strong is too heavy due to the number of injured people. Hence, he contacts the call center and ask them to help fill in the trauma form by communicating with one of the ambulance personnel and to find and prepare some vital information on a specific patient who seems badly injured. When the call center has the requested information ready, Dr. Strong touches the patient's RFID tag, thereby resuming all the data views and forms associated with this patient. The call center highlights specific information on allergies to penicillin and show Dr. Strong some specific sections of particular interest in the record.

Dr. Strong decides to move this patient for hospital treatment immediately and moves the patient to an ambulance. When entering the ambulance, Dr. Strong moves his user session on his wearable computer to the touch display in the ambulance. This display is much larger and hence helps him to have a better overview of the patient's condition. The medical equipment in the ambulance is also used now and all the vital signs from the patients are shown on the display and is transmitted to the call center and on to the hospital. Arriving at the hospital, a staff of well-prepared surgeons and nurses take over the patient who is moved directly to the operating room.

3 Key Design Principles

Based on our research into the working conditions of paramedics and emergency personnel, combined with the a litterature study of the current state-of-the-art within wearable computing, we have identified four key design principles for the Iaso wearable computing platform. These are support for 'preparation and recall', 'focus + context', 'shared collaboration', and 'using multiple displays'. In the following we shall discuss these principles in detail and compare them to related work in this area.

3.1 Preparation and Recall

Most scenarios for wearable computing involve users in situations where there is little time for navigation and information seeking, like train maintenance [16], aircraft inspection [14], the fire service and police force [10], warfare [19], and the emergency workers described in this paper. The key challenge in most of the applications of wearable computing is the limited time and support for information search, retrieval, selection, filtering, and displaying. Based on these observations researchers have proposed to use the principles of 'context-awareness' to have the computer adapt to the user's current work situation [1, 15]. As a supplement to context-awareness in wearable computing we are suggesting 'preparation and recall' as a core design principle for wearable computing. This principle enables a user or one of his colleagues to prepare for a situation and then recall relevant data and/or services when this situation occurs. This is illustrated in the scenario above where the paramedics use a standard template for a patient and recall a prepared map of the accident area.

3.2 Focus + Context

A core challenge in wearable computing is the small screen and resolution of state-of-the-art head mounted displays (HMD)[3]. Furthermore, the standard user-interface mechanisms for managing large amount of information and/or applications on a small screen are not feasible in wearable computing. Overlapping windows, the Windows or Linux taskbar, and 'Alt-Tab' switching between applications is hard to use on a wearable computer. It has been suggested to use principles from Virtual Reality in wearable computing by creating 'Wearable Information Spaces' [5]. The information spaces allow users to navigate between multiple pages of information by moving their head as if they were standing inside a virtual cylinder of information. The same idea has been explored in [13] where they create a virtual space around the user using 3D-sounds. In the emergency scenario such solutions would not work because field workers are constantly moving their head to overlook the scene of the accident. Instead we are pursuing the user-interface design principle of 'Focus + Context' [8, 3], which enables the emergency worker to have an overview of all his activities, over what is taking place within each activity, and then focus on specific aspects within each activity. This is

[3] A typical HMD today is the MicroOptical SV-6 with a VGA resolution on 640x480 or 800x600 in 18 bit color.

illustrated in the scenarios above where the paramedics focus on different medical data while maintaining an overview of his on-going activities.

3.3 Shared Collaboration

Support for collaboration has been identified as central to wearable computing [11]. However, most of the collaboration support pivots around audio and video links, enabling a remote 'expert' to guide the field worker (see e.g. [11, 4, 12]). In many cases, however, the expert *is* the field worker – or he is at least the expert in the current situation. This is typically the case in the emergency scenario. Little attention has been given to enabling the remote user to take active part in the field worker's local activity. Hence, a design principle for collaborative wearable computing is to enable 'activity sharing' in the sense that a remote user can take active part in an activity in the field. This is illustrated in the scenario above where the call center employee searches, sorts, prepares and presents relevant information for the paramedics and where he starts helping the paramedics in filling in the trauma form. In this case the call center employee is participating actively in the work on the site and is not merely providing oral advice.

3.4 Using Multiple Displays

Most research on wearable computing assumes that the only display used is the head mounted display worn by the user. This is, however, seldom the case. In many situations additional displays like PDA, tablet PCs, laptops, and touch screens mounted in e.g. the ambulance are available. A central design principle for wearable computing must be to enable the user to utilize this computational context in his work. For example, to let the paramedic in the scenario use the tablet PC and the screen in the ambulance. This must be achieved with no or minimal overhead – the user must simply be able to walk up to a display and transfer his session. This enables him to utilize the larger screen, resolution, the keyboard and mouse, as illustrated in the scenario above. This also helps users to overcome the limitations in screen size and resolution, and the limited input devices on state-of-the-art wearable computers.

4 Activity-Based Wearable Computing

We are currently working on applying the principles of Activity-Based Computing as a metaphor and platform for wearable computing. Based on the ABC architecture and development framework [2] we have created the Iaso platform for wearable computers. This architecture has special focus on the design principles listed above.

The principles of Activity-Based Computing can be summarized as:

- **Activity-Centered** – A 'Computational Activity' assembles a set of services, needed to support a user carrying out some kind of (work) activity. In the scenario above, there is an activity for each patient which contains services (data view and forms) for (i) the patient medical data, (ii) the trauma form, and (iii) the chart displaying vital signs (see figure 3(b) or figure 2(a)) which illustrate an activity showing these services). This ABC principle supports collecting services into logical bundles around a typical user activity and support the user to easy toggle between activities, thereby supporting alternating work tasks and interruptions.
- **Activity Roaming** – An Activity is stored in an infrastructure and can hence be distributed across a network. An activity can be suspended on one device and resumed on another in another place. This principle supports mobility and enables the user to use available devices nearby – potentially without carrying any device. In the scenario above, activity roaming takes place when Dr. Strong moves his session to the display in the ambulance.
- **Activity Adaptation** – An Activity adapts to the resources available on the device on which it is resumed. Such resources are e.g. the network bandwidth, cpu, or display on a given devices. Hence, an activity might look quite different whether it is resumed on a wall-sized display or on the wearable computer. This principle suggests a path in the middle between thin clients using web services and self-contained applications running on homogeneous devices.
- **Activity Sharing** – An Activity is shared among collaborating users. It has a list of participants who all can access and manipulate the activity. Hence, one user can resume another user's activity and continue working on it. Furthermore, if two or more users resume the same activity at the same time on different devices, they will be notified and if their devices support it, they would engage in an on-line, real-time 'activity sharing' session. This is what takes place in the scenario above where the person at the call center helps Dr. Strong during an activity.
- **Context-Awareness** – An Activity is context-aware, i.e. it is able to adapt and adjust itself according to its usage context. Context-awareness can be used for adapting the user-interface according to the user's current work situation – e.g. by showing medical data for the patient currently in front of Dr. Stong. Or it can be used in a more technical sense, where the execution of

an activity, and its discovery of services, is adjusted to the resources available in its proximity. This happens when Dr. Strong resumes the activity in the ambulance.

4.1 The Iaso Platform

The Iaso platform for wearable computing is illustrated in figure 1. This architecture is built on top of the ABC Framework as a special purpose client tailored for wearable computing. However, many of the core functionalities are directly inherited from the ABC framework, features like support for activity management, storage, activity sharing, context-awareness, collaboration awareness, and the support for activity-adaptation to multiple devices and displays.

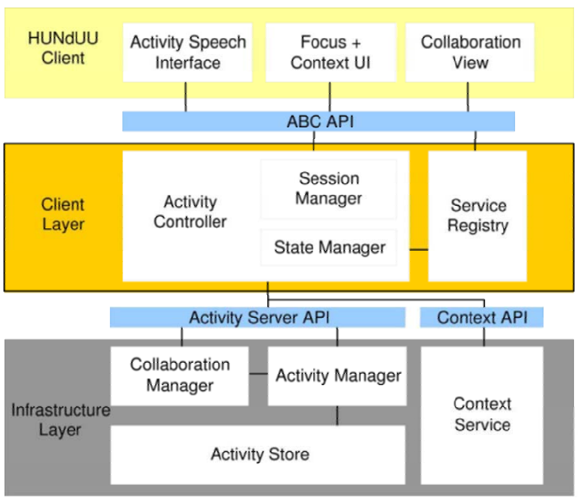

Figure 1. The Iaso system platform as built on top of the ABC architecture.

The ABC Framework consists of an infrastructure layer and a client layer. The infrastructure layer is responsible for storing (the Activity Store) and runtime management of activities (the Activity Manager), for collaborative activity sharing (the Collaboration Manager), and for handling context-information (the Context Service). The infrastructure layer is typically deployed on a set of distributed powerful machines.

The client layer resides on each device participating in the ABC deployment. This layer is a middleware layer between the user-interface and the infrastructure layers. The core component is the Activity Controller, which is responsible for controlling activities on the client and for communicating with the Activity Manager in the infrastructure layer. The State Manager is responsible for handling activity state when an activity is resumed and suspended on various clients. The Session Manager is responsible for handling the collaborative activity sharing session.

The Iaso client resides on top of the ABC framework. It consists of a speech interface client which enables the user to speak to the Activity Controller, thereby enabling him to suspend and resume activities, create new activities, and to navigate within activities. The Iaso client also contains a Focus+Context user-interface component, which helps the user to toggle between an overview of activities and focusing on details within an activity. Finally, the Iaso client contains a list of current participants. The next section described the User-interface of the Iaso client in greater details.

4.2 User Interface

The Focus+Context user-interface component of the Iaso client is illustrated in figure 2 and 3, the former on a wall-sized display and the latter on a wearable computer. Hence, the same activities can be displayed on screen with a substantial difference in screen size and resolution. The first screen on 3(a) is an overview of all active activities for this user, figure 3(b) and figure 2(b) are overviews of a specific activity. Lastly, figure 2(c) and (d) shows details within the activity, focusing on a specific window (or service) in the activity. Each activity consists of a number of applications or services, each running in a separate window. Hence, the trauma form is one application, the map of the accident scene is another.

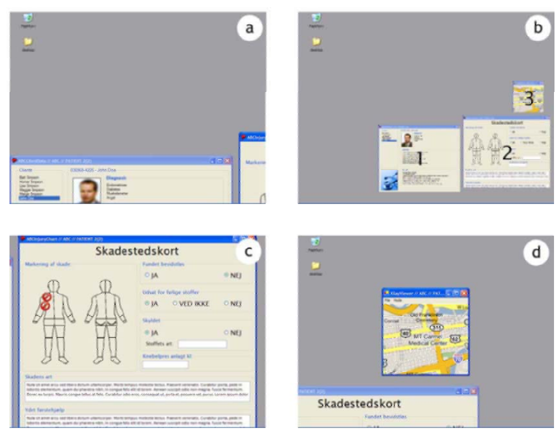

Figure 2. *(a)* **The activity just after it has been resumed.** *(b)* **All services are visible when the user issues a 'Show all services' command.** *(c)* **and** *(d)* **Show two different services (trauma-form and map) which have been 'focused' on.**

The wearable user can switch between these two levels of overview/detail using voice commands. The commands are limited to "Show all services" 2(b), "Focus service `number`" (figure 2(c) and (d)) and "Resume/Suspend activity `activity name`" (figure 2(a) and 3(b)). This is a rather limited set of commands but sufficient to navigate in the activity-based user-interface. Support for voice input in the specific applications (like the trauma form) is not part of the Iaso platform. Hence, support for voice input and navigation within applications and services are specific to the application itself.

4.3 Activity-Based Collaboration

The ABC Framework supports 'Activity Sharing', which means that when two or more users engage in the same activity they have a shared view on the activity and the services or applications running within it. Furthermore, voice links and tele-pointers are created between the participants. In the scenario above, Dr. Strong would e.g. be focusing on the list of injured people on the site (figure 2(b)) while the colleague at the call center is helping him by filling in the patient's trauma form (figure 3(b)). The two users would be able to see what each other is doing and hence work directly together or have a mutual awareness of each other. They would be able to talk to each other and the person in the call center would e.g find some information and present it for Dr. Strong. Dr. Strong would see this information pop up as a new miniaturized window, which he can zoom in on using the voice command. The lay-out of the windows are maintained across platforms. Hence, if the person at the call centers uses a monitor with a 1680x1050 resolution while collaborating with Dr. Strong using the wearable monitor with a 640x480 or 800x600 resolution, they will still see the same layout of windows within the activity, but at different zoom levels. Hence, the activity view (figure 2(a)) shows the whole activity zoomed to the current display.

The degree of collaboration within each application is dependent on the application itself. Hence, the degree of collaboration support in the 'Trauma Form' application depends on its implementation as some kind of client-server system where both Dr. Strong and the person at the call center can access shared data (a web service for example). Hence, the Iaso platform enables collaboration on the activity level and cannot guarantee collaboration within each service[4].

4.4 Context-Aware Activity Retrieval

Creating a temporary but unique identification of more or less unconscious patients is, according to the emergency workers, a central challenge today. Often, patients are identified by their traits and type of injury, which is very imprecise. The Iaso platform is context-aware and allows for context-based retrieval of an activity. Hence, an activity can be linked to contextual information. This feature is exploited to link simple RFID tags attached to the patient with the patients activity and hence his medical and personal data. By scanning the patient's RFID tag, the Iaso clients present this patient's activity with all its relevant information.

Context-awareness is also used to help Dr. Strong move his session from the wearable to the display in the ambulance. When he and the patient enter the ambulance, Dr. Strong is able to resume his activity on this display.

5 Implementation and Deployment

The platform has been implemented in a version 1 on top of version 4 of the ABC Framework, where the infrastructure layer is implemented in Java and the client layer implemented on the Microsoft .NET platform in C#. The speech interface has been implemented using the Microsoft Speech SDK 5.1. The prototype has been evaluated by members of our research team and has been improved based on their feedback.

The Activity Manager supports both a request-response access to stored activities as well as an event-based subscribe-publish-notify mechamism for distributing events. We have designed an ABC Protocol (ABCP) which supports clients to request and store activities, and to subscribe to changes in activities. An Activity Markup Language (AML) is used to describe activities and events. The ABC Protocol runs over TCP/IP sockets and hence provides and language independent implementation between the infrastructure layer and the client layer.

The .NET client provides the middleware-layer on which the applications run. This layer manages the state of all applications and furthermore monitors the client computer for any changes in the state of the current activity, e.g. a new application being started. This is all done transparently to the user. We have implemented wrapper classes which make it possible to use existing applications such as Internet Explorer, Word, and Notepad. We also provide an API which may be used for creating new Iaso-aware applications. Creating a new application is quite simple; simply extend the class and provide implementations of a state-setting and state-getting method.

Regarding deployment, our current design pivots around a centralized infrastructure layer with one or more coordinating activity servers and a wearable client running Windows XP with the ABC framework and the Iaso software. This is in many ways a 'heavy' platform for wearable computing, which requires substantial hardware and networking

[4]The ABC Framework has an API that helps programmers to develop or extend existing application to make them 'ABC-aware', including participating in activity sharing (see [2]). However, the ABC platform also supports legacy applications without modifications, but in this case we cannot guarantee collaboration support.

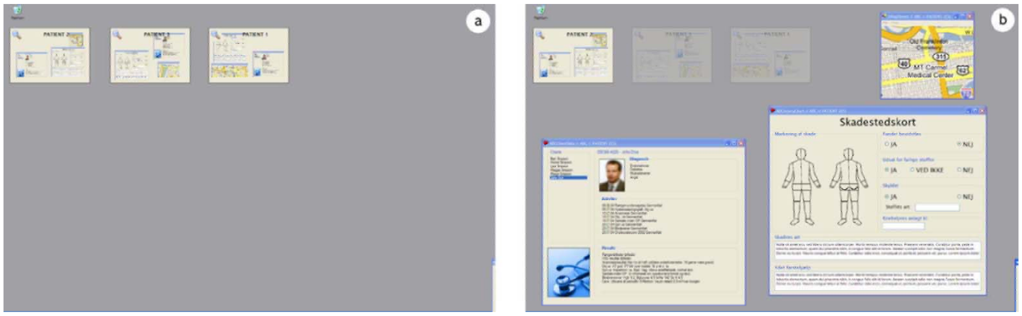

Figure 3. *(a)* **An overview of all activities.** *(b)* **A single activity has been resumed.**

resources. However, our initial research into the operating conditions of the accident scenario makes us confident that these deployment requirements can be meet. The wearable unit can be deployed on the Xybernaut MA-V, which runs Windows XP and the infrastructure layer can run on small servers installed in the ambulances and other vehicles arriving at the scene. These vehicles can also provide WLAN coverage for the accident scene and can communicate back to the call center using UMTS. It should be noted that the requirement for network bandwidth from the event-based activity sharing in the ABC Framework is rather limited, even in collaboration mode. Hence, the largest demand for bandwidth would emerge from the use of audio and/or video.

6 Conclusion

The main contribution of this paper has been to introduce four design principles for wearable computing, which supplement the work within this research area. These principles come out of our design of wearable computing for emergency situations, and are:

- Preparation and Recall
- Focus + Context
- Shared collaboration
- Using multiple displays

We have presented the Iaso platform for wearable computing, which seeks to address these principles. This platform was built on top of the ABC Framework and hence supports 'Activity-Based Wearable Computing'. We believe that this platform meets the four design principles above.

Firstly, the design principles of 'Preparation and Recall' is evident in the Iaso platform by enabling the user to prepare various activties and store them for later retrieval. This preparation can take place on other computer devices than the wearable one, thereby enabling the user to take advantage of keyboard, mouse, and other I/O devices during preparation. By organizing computational services in activities, which are tailored to expected future work activities on the accident site, the emergency workers are able to have resources ready and to easy toggle between them when arriving at the site.

Secondly, the Iaso platform contains a novel, custom designed user-interface which helps the user to get an overview of all his running activities as well as focusing on specific details within these activities. By zooming in on a specific service and zooming out to see a whole activity or the whole set of activities, the user-interface seeks to support the design principle of 'Focus + Context'. For example, the user can by simple voice commands toggle between a large overview of all the activities and detailed information on e.g. the patient or a map of the accident site.

Thirdly, the Iaso platform enables the users (on site as well as persons providing back-up) to collaborate equally and help each other to carry out an activity. This 'Shared Collaboration' is basic to the ABC Framework and is hence a central part of the Iaso platform. The support for activity sharing enables users to join activities and users sitting at e.g. the emergency call center or physicians at the hospital is able to join an activity on the accident site and help the on-site user by finding and displaying information, or to engage directly in the activity by e.g. entering data. The current version of the Iaso platform does not support video communication only audio links. Video is an important part of a final deployable wearable computer support for emergency work and we are currently investigating the best design of video use. One suggestion is to have head mounted cameras (as in [11]). Another option is to have fixed cameras on-site mounted e.g. on ambulances and other vehicles. However, focus of this paper has not been on video cooperation, but the support for sharing data and views with other mobile and non-mobile collegues. This 'Activity Sharing' is a different approach to collaboration than the 'expert-helps-novice-using-video' approach seen so far in wearable computing.

Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)
1545-0678/05 $20.00

Finally, by using the ABC Framework the Iaso platform enables the user to utilize computational devices in the nearby context. We can probably all agree that wearables are difficult to use (the Twiddler keyboard has a steep learning curve) and the support for moving whatever you are doing to a 'real' computer is not trivial. Hence, support for moving the users session along with all the user's activities to the computer in the ambulance is essential.

In the future we plan to continue this work and we are in the process of setting up both in-house and on site evaluation session with emergency workers. Currently, researchers from our research team is doing on-site ethnographic studies of emergency work which will help us design the type of application to run on the Iaso platform as well as provide some more feedback on the design of the platform itself. At a later point, we plan to evaluate the support for wearable computing during emergency drills where everything but the injured patients are for real.

Acknowledgments

The research presented in this paper has been funded by ISIS Katrinebjerg competence center, Aarhus, Denmark. The design ideas in the bus acident scenario were developed by the students attending the 'Designing Interactive Systems (DIS)' class at the University of Arhus in the Fall 2003 taught by Morten Kyng and Preben Mogensen.

References

[1] G. D. Abowd, A. K. Dey, R. Orr, and J. Brotherton. Context-awareness in wearable and ubiquitous computing. In *Proceedings of the 1st IEEE International Symposium on Wearable Computers*, page 179. IEEE Computer Society, 1997.

[2] J. E. Bardram. Activity-Based Support for Mobility and Collaboration in Ubiquitous Computing. In L. Baresi, editor, *Proceedings of the Second International Conference on Ubiquitous Mobile Information and Collaboration Systems (UMICS 2004)*, Lecture Notes in Computer Science, pages 169–184, Riga, Latvia, Sept. 2004. Springer Verlag.

[3] P. Baudisch, N. Good, V. Bellotti, and P. Schraedley. Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 259–266. ACM Press, 2002.

[4] M. Bauer, T. Heiber, G. Kortuem, and Z. Segall. A collaborative wearable system with remote sensing. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, page 10. IEEE Computer Society, 1998.

[5] M. Billinghurst, J. Bowskill, M. Jessop, and J. Morphett. A wearable spatial conferencing space. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, page 76. IEEE Computer Society, 1998.

[6] H. B. Christensen and J. E. Bardram. Supporting Human Activities – Exploring Activity-Centered Computing. In G. Borriello and L. E. Holmquist, editors, *Proceedings of Ubicomp 2002: Ubiquitous Computing*, volume 2498 of *Lecture Notes in Computer Science*, pages 107–116, Göteborg, Sweden, Sept. 2002. Springer Verlag.

[7] R. DeVaul, M. Sung, J. Gips, and A. S. Pentland. Mithril 2003: Applications and architecture. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, page 4. IEEE Computer Society, 2003.

[8] S. Feiner and A. Shamash. Hybrid user interfaces: breeding virtually bigger interfaces for physically smaller computers. In *Proceedings of the 4th annual ACM symposium on User interface software and technology*, pages 9–17. ACM Press, 1991.

[9] S. Finger, M. Terk, E. Subrahamanian, C. Kasabach, F. Prinz, D. P. Siewiorek, A. Smailagic, J. Stivoric, and L. Weiss. Rapid design and manufacture of wearable computers. *Communications of the ACM*, 39(2):63–70, 1996.

[10] D. J. Haniff and C. Baber. Wearable computers for the fire service and police force: Technological and human factors. In *Proceedings of the 3rd IEEE International Symposium on Wearable Computers*, page 185. IEEE Computer Society, 1999.

[11] G. Kortuem, M. Bauer, and Z. Segall. Netman: the design of a collaborative wearable computer system. *Mobile Network and Applications*, 4(1):49–58, 1999.

[12] R. E. Kraut, M. D. Miller, and J. Siegel. Collaboration in performance of physical tasks: effects on outcomes and communication. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pages 57–66. ACM Press, 1996.

[13] J. Lumsden and S. Brewster. A paradigm shift: alternative interaction techniques for use with mobile & wearable devices. In *Proceedings of the 2003 conference of the Centre for Advanced Studies conference on Collaborative research*, pages 197–210. IBM Press, 2003.

[14] J. J. Ockerman and A. R. Pritchett. Preliminary investigation of wearable computers for task guidance in aircraft inspection. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, page 33. IEEE Computer Society, 1998.

[15] J. Pascoe. Adding generic contextual capabilities to wearable computers. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, page 92. IEEE Computer Society, 1998.

[16] D. Siewiorek, A. Smailagic, L. Bass, J. Siegel, R. Martin, and B. Bennington. Adtranz: A mobile computing system for maintenance and collaboration. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, page 25. IEEE Computer Society, 1998.

[17] A. Smailagic and D. P. Siewiorek. A case study in embedded-system design: The vuman 2 wearable computer. *IEEE Design and Test*, 10(3):56–67, 1993.

[18] J. P. Sousa and D. Garlan. Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments. In *Proceeding of the 3rd Working IEEE/IFIP Conference on Software Architecture*, Montreal, August 25-31 2002.

[19] M. J. Zieniewicz, D. C. Johnson, D. C. Wong, and J. D. Flatt. The Evolution of Army Wearable Computers. *IEEE Pervasive Computing*, 1(4):30–40, Oct. 2002.