

A Collaborative Wearable System with Remote Sensing

Martin Bauer, Timo Heiber, Gerd Kortuem, Zary Segall
Computer Science Department
University of Oregon
Eugene, OR 97403, USA

{mabauer,timo,kortuem,zs}@cs.uoregon.edu

Abstract

This paper presents a collaborative wearable system based on the notion of remote sensing. Remote sensing lets users of wearable or stationary computers perceive a remote environment through the sensors of a remote wearable computer. We describe a concrete system with remote sensing capability that is designed to enhance the communication and cooperation of highly mobile computer technicians.

1: Introduction

There is an obvious need for effective communication and collaboration in typical wearable domains such as maintenance, repair, construction and manufacturing. As noted by Siegel et al [20], having wearable computer systems that allow field workers to access information and contact experts can be valuable in many settings, from airline maintenance, to health care, emergency response and on-the-job training.

In this paper we report on the design and implementation of a collaborative wearable computer system that supports computer technicians in their task of maintaining a campus-wide computer network. The wearable computer is a Personal Computer-like device with a head-mounted display that integrates video camera, microphone and speaker. The computer is worn by a technician in the pouch of a specially designed vest (see Figure 1).

The wearable system, which we call NETMAN, enables technicians in the field and office-based experts to collaborate in real-time using audio and video. A camera, that is attached to the wearable computer and points away from the user at the task area, enables a remote expert to see what the technician in the field sees and direct his or her attention using a remote pointer.

While similar collaborative wearable systems have been proposed before or are currently under development [5;6;15;17], the NETMAN system uses an innovative approach to enhance and enrich the collaboration between

remote users, based on the concept of *remote sensing*. Remote sensing means that a remote user, the expert sitting at a desk in an office or another wearable user, has direct, unmediated access to output of sensors attached to another user's wearable computer. This concept is visualized in Figure 2.

The term 'remote sensing' originally describes the process of measuring and analyzing electromagnetic radiation for the purpose of understanding and managing the Earth's resources and environment [18]. It has also been used in the context of telepresence and telerobotics systems (e.g., [12;19]). For our purpose we define remote sensing more broadly as the collection of information about an object without being in physical contact with the object.

Remote sensing requires sensors such as cameras, microphones, laser scanners, and receivers for radio or infrared radiation. Such sensors have long been used in wearable computing to provide users of a wearable device with an enhanced view of the immediate environment. The NETMAN system allows two or more users to share this enhanced view by transmitting sensory data to remote computers over a wireless network. Depending on the number and type of sensors, this approach enables remote

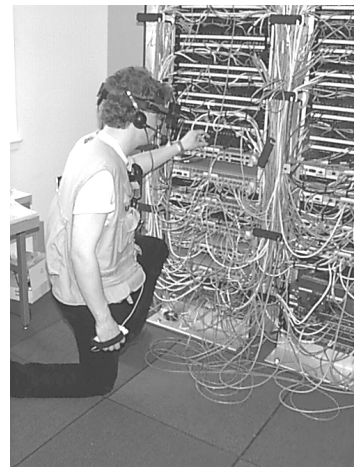


Figure 1: NETMAN Wearable Computer

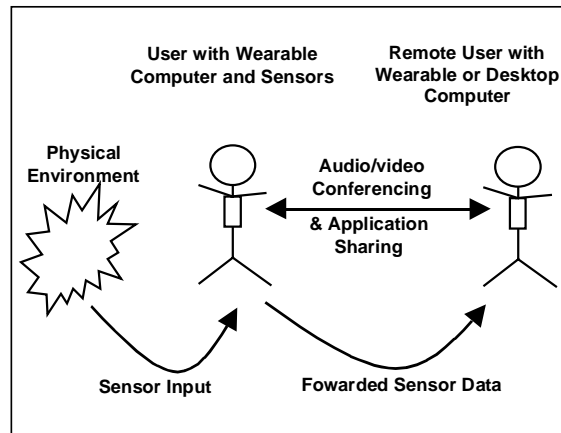


Figure 2: Remote Sensing

users to perceive a remote environment almost as if they were physically present.

The idea of using remote sensing as a way to enhance collaboration was motivated by our experience with earlier prototypes of NETMAN [14]. We soon realized that the quality of the video image that can be achieved in a wearable setting tends to be rather poor. Limitations of the wireless network in terms of bandwidth and delay characteristics restrict the usable image resolution and frame rate. Small screen size and poor quality of current head-mounted displays directly effects the perceived image quality. This together with poor lighting conditions can make it impossible for a remote user to identify any significant details in the transmitted picture.

Remote sensing is a natural generalization of shared audio/video capabilities found in ‘traditional’ collaborative wearable systems. It is an attempt to overcome limitations of such systems by providing a remote expert with additional (besides video and audio) and more accurate information about the state of the environment and what exactly the technician is doing.

The remainder of this paper is structured as follows. In the next section we discuss related research and introduce the application scenario that underlies the design of NETMAN. In Section 3, we discuss general design considerations for collaborative wearable systems. Section 4 outlines the NETMAN prototype, while Section 5 describes remote sensing applications and the underlying software infrastructure. In Section 6, we discuss early experiences with the prototype. Section 7, finally, summarizes.

2: Collaborative Wearable Systems

Most wearable computers today are designed as stand-alone systems that provide users with automatic, context-

sensitive access to information, but do not support inter-personal communication and collaboration. In a similar vein, previous work on collaborative systems has almost exclusively focused on white-collar workers in office settings. Communication needs of mobile field workers, whose work includes a high amount of manual activities, such as technicians and repair personnel, were mostly ignored.

Recent research suggests that for certain domains collaborative wearable systems with shared audio/video capabilities can have a positive effect on workers’ collaboration and coordination:

In [15;20] the authors report the results of two CMU studies on mobile collaborative systems for the support of maintenance task of bicycles and aircrafts. They describe a wearable system for collaboration with a remote expert using shared video and hypertext: “Preliminary results suggest that doing the tasks with a more experienced helper with shared video to support coordination of work is the most effective treatment. Sharing a view of the work area helped the team to coordinate their conversation”.

Similar research was performed at the University of Washington [5;6]. The authors describe two pilot studies, which imply that wearables may be able to support three-dimensional collaboration and that users will perform better with these interfaces than with immersive collaborative environments.

Finally, Boeing is currently investigating the use of wearable video conferencing systems for fast and accurate communication of airplane mechanics at remote locations and/or mechanics working on different parts of the same airplane [17].

None of these systems, however, makes advanced use of sensors. They are mainly mobile videoconference systems that solely rely on audio and video signals to support collaboration. Yet one of the most interesting and most novel aspects of wearable computing is the combination of sensors and contextual awareness. Using sensors such as proximity sensors, location sensors, and electronic tags for identification of nearby objects, a wearable computer can actively gather knowledge and information about its environment and use it for advanced automatic and context-sensitive support of users. Examples include: context-sensitive user interfaces, context-based reminder systems, and context-based retrieval systems [1;2;3;4;10]. We believe that in collaborative settings remote participants can benefit in similar ways from having direct and unmediated access to another user’s sensory data.

We decided to test our ideas about remote sensing and collaborative wearable systems using a real-world application with hard requirements. In the following section we will introduce the application scenario that underlies the development of NETMAN.

2.1: Application Scenario

Our goal for the NETMAN project was to design and develop a wearable system that helps technicians in their daily task of troubleshooting and repairing faults in computer network equipment. For collecting requirements we are working closely with the University of Oregon Computing Center which is responsible for maintaining the computer and network installations throughout campus. Typical tasks of technicians include: installation of new network equipment such as routers; performing regularly scheduled maintenance work; troubleshooting of network faults; repair and replacement of faulty equipment.

The technicians who are sent out to locate and, if possible, resolve network problems are equipped with an array of communication devices like cellular phone, walkie-talkie, pager, and - in some cases - a notebook computer. Skills and experiences of field technicians vary and can range from inexperienced student volunteers to highly trained experts. In most cases, however, technical knowledge of technicians is limited, but sufficient to perform routine repairs. As part of their work, technicians often have to perform manual activities like opening computers, moving furniture and equipment, dragging wires, crawling under desks, suggesting a wearable computer design with hands-free operation.

In addition to field technicians, the Computing Center employs a limited number of full-time employees who are

knowledgeable experts in their domain. For the most part, they rarely leave their office and do not perform routine repairs.

Field technicians often have to visit a particular site several times before they are able to resolve a problem, because they need to look up information, get additional equipment, or ask a more experienced technician for advice. For example, they might call into the office to ask questions like “*How do I do...?*”. Depending on the expertise of the technician, extensive communication between field technicians and experts at the office can be required to solve a particular problem. A wearable audio/video conferencing solution clearly could be helpful in this context: the expert is able to answer questions more quickly or more accurately if he or she is able to see the remote work area and what the technician is working on.

To our surprise we found that the most frequent cause for network problems is not related to hardware faults or software configuration issues, but misconnected wires. This type of problem can very efficiently be resolved by two closely cooperating technicians: the expert can perform tests and analyze the status of the network from the office, while the technician in the field rewires cables at a network closet or other computer devices.

Through experiments with earlier prototypes of NETMAN we made the observation that the quality of the video image can severely effect collaboration. For example, Figure 3. shows the video image as seen on the remote expert’s computer. The video quality in terms of resolution, frame rate, and delay characteristics is good enough to allow the remote expert to determine where a wearable user is, in which direction he or she is looking in, and what objects are in the environment. Yet, it is impossible to clearly identify details like shape and type of connectors or read printed labels. This information is necessary for the expert to determine if cables are connected properly.

This observation lead us to the idea to use additional sensors, most notably sensors for object identity and location, in order to provide the remote expert with an enhanced view of the technician’s work area.

3: Design Considerations

During the design phase of NETMAN we studied a number of possible design alternatives. We finally came up with a list of six possible *collaboration functions* from which a collaborative system can be assembled. These collaboration functions define a design space for (synchronous) collaborative wearable systems:

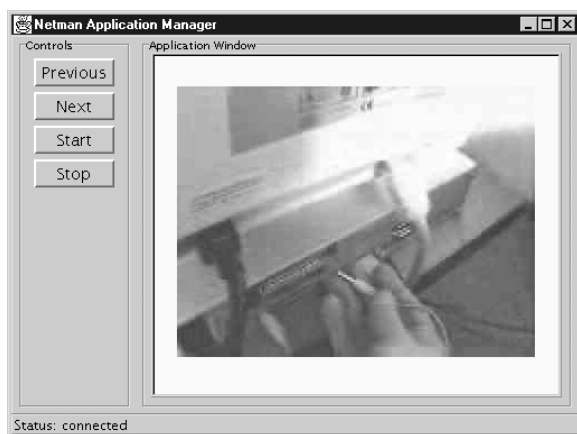


Figure 3: Remote Video Image

- 1) *Remote awareness*: It has been shown in the CSCW literature that users of collaborative systems feel more comfortable when they know who else participates in a conversation [8;9;11]. Awareness of remote communication partners can be achieved in many ways, for example, by presenting icons or pictures of each participant. For wearable applications, one can also think of audio-only representations.
- 2) *Remote presence*: Going one step beyond remote awareness, remote presence provides a richer and more natural conversation by using live representations of participants. This can be in the form of a live-feed from a camera showing a user's face or in form of an avatar, a 3-dimensional representation of a user's face or body controlled by a remote user. In both cases, remote presence gives users the ability to convey non-verbal clues using gestures and facial expressions, resulting in an improved intimacy between communication partners and a feeling of co-presence.
- 3) *Remote presentation*: By remote presentation we mean a user's ability to superimpose images over a wearable user's (real-world) view. By using shared computer screens it is possible for a participant to put a wiring diagram in the field of view of another wearable user. Remote presentation is thus an effective means for sharing information and focusing verbal communication.
- 4) *Remote pointing*: The ability to control a remote cursor enables users to point at objects in other users' view. Such objects can either be virtual objects (a wire in a wiring diagram) or real-world objects captured by the camera of a wearable computer. Like remote presentation, remote pointing can increase the effectiveness of verbal communications by directing the participants' attention.
- 5) *Remote sensing*: Remote sensing means that a remote user has direct, unmediated access to output of sensors attached to another user's wearable computer. Remote Sensing has the potential of streamlining the conversation among several collaboration partners by helping them to establish a shared conversational context and by creating a heightened sense of copresence. For example, participants do not have to talk explicitly about which computer one of them is standing in front of, because this information is available automatically to each participant. Remote sensing allows users to perceive a remote environment almost as if they were physically present.

- 5) *Remote manipulation*: Remote manipulation, finally, goes beyond remote sensing and refers to a user's ability to manipulate objects in another user's physical environment.

Collaborative wearable systems discussed in the literature ([5;6;15;20]) focus almost exclusively on remote presentation and remote pointing. NETMAN goes beyond these systems by adding remote sensing as a third component. In this paper we focus on the remote sensing aspect of NETMAN, while remote presentation and remote pointing are addressed in [14].

4: The NETMAN Prototype

The NETMAN system is a distributed groupware system that consists of several hardware and software components (Figure 4):

- One or more wearable computers worn by field technicians during repair and maintenance tasks;
- various sensors attached to wearable computers;
- one or more desk bound workstations used by expert technicians in offices at the Computing Center;
- application and system software running on both the wearable computer and the workstations;
- a central database server that stores information about computer and network equipment found throughout campus.

We will now describe each of these components in detail.

4.1: Wearable Computer

The wearable computer we use in NETMAN is based upon a Pentium motherboard from Texas Instrument and runs Windows95 as operating system. The computer is housed in a specially designed vest that accommodates the various batteries and input devices (Figure 1). The central processing unit is fitted into a pouch on the back, and cables are run from the CPU out to the front pockets in the vest. These cables feed the batteries and input devices positioned in the front of the vest. The weight of the batteries and accessories counters the weight of the CPU pouch on the back, providing a comfortable fit. A head-mounted display is used for output. The primary form of user input is keyboard input using a Twiddler keyboard. More details on the design of the wearable computer can be found in [10;13;14].

We are also experimenting with a commercial wearable computer, the FlexiPC by VIA Corporation. This computer combines a lightweight design with easy

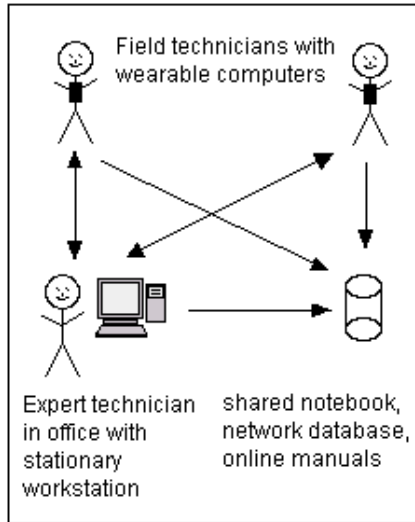


Figure 4: NETMAN Overview

extensibility and features a hand-held display and pen-input. While this combination is not a wearable computer in the strict sense, because it doesn't provide hands-free operation, we use it as our primary development and test platform.

All wearable computers and stationary workstations are directly connected to the Internet. We are using a Metricom wireless local-area, which covers the entire University of Oregon campus.

4.2: Sensors

Each wearable device is equipped with the following sensors for location, object identity, and analyzing network traffic:

Location: The first sensor is an infrared receiver for determining the wearable's location inside buildings. The IR sensor receives signals sent out by IR transmitters that are attached to the ceiling of various rooms in our lab. Each transmitter broadcasts a unique signal allowing the wearable to look up its location in a centralized database. The IR receivers and transmitters are based on a proprietary design and are described in more detail in [13;14].

Object Identity: The second sensor is a scanner for electronic equipment tags. We use iButtons from Dallas Semiconductor [7] as electronic tags for equipment, and iButton scanners (so-called 'Blue Dot Receptor') as sensors. An iButton is a 16mm computer chip housed in a stainless steel case. Each iButton has a unique, unalterable, 64-bit unique registration number stored on the silicon chip that we use to uniquely identify computer

equipment.¹ By attaching iButtons to computers, routers, network outlets, and even individual wires we are able to uniquely identify objects that are important to network technicians. In order to read the registration number of an iButton, the user touches the button with the iButton scanner, which is connected to the parallel port of the wearable computer. A daemon process, written in Java using the iButton development kit, runs on the wearable device and listens to signals coming from the iButton scanner.

A centralized database stores information about iButton-enhanced objects. Currently, the only information stored about objects is their location and type, that is, whether an object is a computer, a router or something else. At a later time we plan on storing this information directly in the memory of iButtons (iButtons with up to 4K memory are available), which would eliminate the need for a centralized database.

Network Traffic: The third sensor is a packet sniffer, a device that plugs into network outlets and allows technicians to analyze network packets.²

4.3: Software Applications

Several applications make use of the information delivered by these sensors. Among them are:

- a context sensitive *document browser* that uses the input of the iButton scanner to automatically search a database for documents about the equipment the technician is working on;
- an *interactive map* that displays a building floor plan showing the location of various types of computer and network equipment, such as routers and network outlets (Figure 5). This application is used by technician and expert to identify which piece of equipment they are talking about, and to access information about network equipment by location (see below);
- a *network analyzer*, a software packet that analyzes and visualizes the information about the network traffic delivered by the packet sniffer.

Because of the characteristics and limitations of the input devices of the wearable computer we have abandoned some features typical of current GUI interfaces, most notably the desktop-metaphor and the concept of movable

¹ We use DS1990A iButtons with 64-Bit ROM.

² The packet sniffer is not implemented in the current prototype.

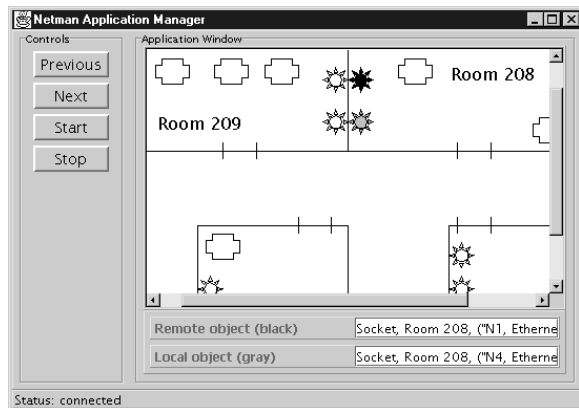


Figure 5: Interactive Map Application

and resizable windows. Both concepts have proven very successful for desktop computers, but seem inappropriate for wearable computers with limited screen space and restricted input device options.

The desktop has been replaced by the so-called *Application Manager*, which allows the user to toggle between application modules. The application manager provides a streamlined user-interface for switching among several application modules. Using the *Previous* and *Next* buttons visible in Figures 3 and 5, the user can switch from application to application by simply pressing one button. While several applications can be running at the same time, only one applications is visible in the main window of the application manager.

Each application module is an independent software entity that plugs into the *Application Manager*. The *Application Manager* provides a simple API that makes it easy for software developers to write new application modules.

5: Remote Sensing

All three applications mentioned above make use of remote sensing. This means that, while copies of each application are running on the wearable computer as well as on the workstation of the remote expert, sensor input is simultaneously sent to the applications on both computers (the exact mechanism is explained below).

To explain how remote sensing works let's assume that both users have decided to work together to resolve a particular network problem. Let's further assume that they have decided to check the wiring in a building.

Through the camera of the wearable computer the expert can see the remote work area, albeit in low quality. As they talk about how equipment is connected or how it should be connected, they use the map application to

indicate what device or network outlet they are referring to (Figure 5).

The map application is a shared-window application, so that both users see identical screens. The symbols in the map represent various types of network equipment. For example, network wall sockets are indicated by stars. Both users can select symbols to indicate which particular piece of equipment they are referring to. Objects can be selected by either scanning the iButton tag attached to a device in the real world, or by clicking on the symbol on the screen. The selections of the local and remote user are indicated by colors: the object that was selected or scanned by the local user is indicated by the color gray, whereas the selection of the remote user is displayed in black. Additional information about the selected devices is displayed in the text fields at the bottom of the screen.

When asked how a particular router is connected to the network, the technician scans the electronic tag of the respective outlet saying: "It is connected to *this* outlet." This action highlights the outlet in the map application so that both users see what outlet the technician is referring to. Conversely, when asking "And what is connected to *this* socket?" the expert selects an outlet on the map. Seeing which object the expert selected, the technician then scans several sockets till he finds the one indicated by the expert. On the map the technician can observe which piece of equipment the expert has selected and which he himself scanned last. As response the technician then scans the tag of the connected device. This way the expert not only knows what type of device is connected to this outlet, but using the *document browser* has immediate access to all the relevant information without having to manually look for it.

The network analyzer software could be used in a similar vein. Asking the technician to plug the network sensor into a specific outlet (again indicating it on the map) the expert could then analyze the network traffic at this particular outlet from his workstation.

This description shows how remote sensing can facilitate collaboration by helping to disambiguate the meaning of pronouns or verbal descriptions like "the router over there". Again, the quality of the video image makes it impossible to use a shared view for a similar purpose.

5.1: Sensor Forwarding

Remote sensing is implemented using a set of specially designed system services. The overall software organization of the wearable device follows a layered architecture as shown in Figure 6.

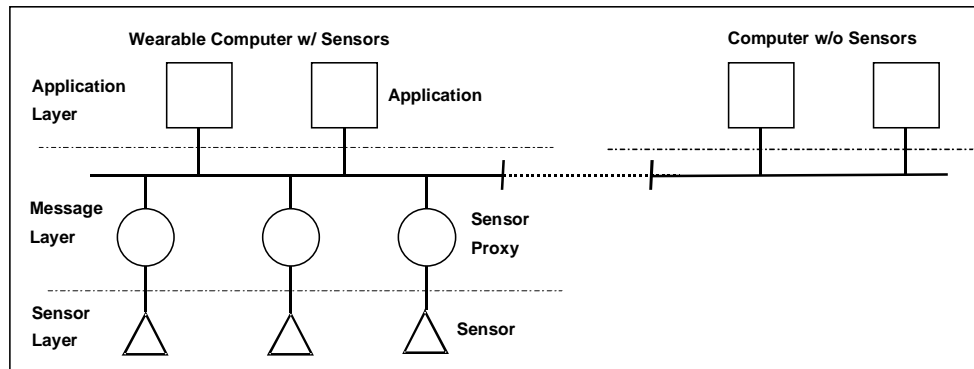


Figure 6: Software Infrastructure for Sensor Forwarding

The top layer consists of several software applications as described above. Application modules receive input from one or more sensors using an event-based communication mechanism. An event-bus connects application modules with so-called *sensor proxies*³. Sensor proxies are background applications (daemons) that provide a unified, event-based API to heterogeneous sensors and make it easy for applications to talk to a wide variety of sensors. This approach enables us to easily integrate new types of sensors, or to replace one sensor with another one of the same type. In order to receive sensor data from a particular sensor, applications register with the event bus:

```
int register(<host-name>, <sensor-type>)
```

Upon registration applications receive notification events through the event-bus whenever a new sensor reading is available. For example, the call

```
register("localhost", "iButton")
```

causes the calling application to be notified whenever the iButton reader scans a new iButton.

In general, each application can register with several sensors and each sensor can have several applications it sends input to. The event mechanism is implemented using Sun's InfoBus architecture [21].

In our prototype, the workstation of the office-based expert runs essentially the same software as the wearable computer⁴. However, since (in our scenario) there are no sensors attached to the stationary workstation, the bottom two layers are missing. Application modules on the workstation can register with sensor proxies on the

wearable computer just like local applications by specifying the host name of the wearable computer.

The distributed event-bus connects wearable and stationary computer (or two or more wearable computer) seamlessly, and ensures that sensor data are transparently forwarded to applications on the remote machine.

5.2: Application Sharing

In addition to the application modules described above, we realized a number of shared applications, which simultaneously run on both participants' machines. These applications realize more traditional forms of synchronous collaboration (remote presentation and remote pointing).

These modules are: (1) a shared *web browser* for accessing online manuals, help files, configuration files etc. which are stored on a central LAN server; (2) a shared *video viewer* which displays the current image of the wearable camera (Figure 3).

6: Discussion

The current NETMAN system is an early prototype, which has not yet seen formal evaluation or deployment in the real world. Preliminary observation point to the validity of some of the technical solutions employed in NETMAN. In particular the sensor-proxy approach has proved to provide a useful level of abstraction. It facilitates the construction of remote-sensing applications in two ways: (1) applications do not need to be concerned about characteristics of individual sensor types. Sensor proxies provide applications with a unified view of sensory input, whether the input comes from a GPS, an IR sensor or an iButton scanner. This fact makes the design of applications much simpler. We anticipate that this concept will also be useful for the design of traditional wearable systems without remote sensing. (2) Sensor-proxies allow us to switch the implementation of a particular sensor type

³ The term 'sensor proxy' was introduced by Ullmer and Ishii in [22].

⁴ In a more sophisticated prototype the client software running on the stationary workstation could use a more traditional user-interface with multiple independent windows.

without effecting the application. For example, in the future we could easily switch from using iButtons for object identification to wireless electronic tags.

The user-Interface of the current prototype is traditional compared to other approaches. For example, MacIntyre [16] describes an audio-based augmented-reality interface to intelligent environments. However, we believe that even simple interfaces in connection with remote-sensing can provide a significant advantage over systems without remote sensing.

7: Conclusion

The combination of wearable computing and remote sensing introduces new and interesting ways of interacting with the real world. By creating a rich shared conversational context, remote sensing has the potential of significantly enhancing the collaboration of remote users. Wearable computers provide a sole remote-sensing platform, because of their unique combination of mobility, perception, and context-awareness.

In this paper we have described a concrete system with remote sensing capability. In particular we have shown an architecture for forwarding sensory data between wearable computers and how this architecture can be used to implement remote sensing applications. Furthermore, we have shown a real-world usage scenario for wearable remote sensing. The described applications are simple, yet we believe provide a glimpse of the future potential of wearable remote sensing systems. More sophisticated systems will include different and advanced types of sensors and more sophisticated user-interfaces. In future work we hope to apply remote sensing to other domains and integrate additional types of sensors.

8: Reference List

1. Abowd, G. D.; Atkeson, Ch. G.; Hong, J.; Long, S.; Kooper, R., and Pinkerton, M. Cyberguide: A Mobile Context-Aware Tour Guide. Baltzer/ACM Wireless Networks. 1997; 3.
2. Abowd, G. D.; Dey, A. K.; Orr, R., and Brotherton, J. Context-awareness in Wearable and Ubiquitous Computing. First International Symposium on Wearable Computing; 1997.
3. Beadle, H. W. P.; Harper, B.; Maguire Jr., G. Q., and Judge, J. Location Aware Mobile Computing. IEEE/IEE International Conference on Telecommunications, (ICT'97); 1997 Apr; Melbourne.
4. Beadle, H. W. P.; Maguire Jr., G. Q., and Smith, M. T. Using Location and Environment Awareness in Mobile Communications. EEE/IEEE ICICS'97; 1997 Sep; Singapore.
5. Billinghamurst, M.; Bowskill, J.; Dyer, N., and Morphet, J. An Evaluation of Wearable Information Spaces. VRAIS '98.
6. Billinghamurst, M.; Weghorst, S., and Furness, T. A. Wearable Computers for Three-Dimensional CSCW. First International Symposium on Wearable Computing; 1997; Boston, MA.
7. Dallas Semiconductor. <http://www.ibutton.com>.
8. Dourish, P. and Bellotti, V. Awareness and Coordination in Shared Workspaces. Proceedings CSCW'92; 1992.
9. Dourish, P. and Bly, S. Portholes: Supporting Awareness in Distributed Workgroup. Proceedings CHI'92;
10. Fickas, S.; Kortuem, G., and Segall, Z., Software Organization for Dynamic and Adaptable Wearable Systems. First International Symposium on Wearable Computing; 1997; Boston, MA.
11. Gutwin, C.; Stark, G., and Greenberg, S. Supporting Workspace Awareness in Educational Groupware. Proceedings CSCW'95; 1995; Bloomington.
12. Kontarinis, D. A. and Howe, R. D., Tactile Display of Vibratory Information in Teleoperation and Virtual Environments. Presence: Teleoperators and Virtual Environments. 1995 Fall; 4(4):387-402 .
13. Kortuem, G.; Segall, Z., and Bauer, M., Context-Aware, Adaptive Wearable Computers as Remote Interfaces to 'Intelligent' Environments. Second International Symposium on Wearable Computing; 1998; Pittsburgh, PA.
14. Kortuem, G.; Segall, Z.; Bauer, M., and Heiber, T. NETMAN: The Design of a Collaborative Wearable Computer System. MONET: Mobile Networks and Applications. to appear.
15. Kraut, R. E.; Miller, M. D., and Siegel, J. Collaboration in Performance of Physical Tasks: Effects on Outcome and Communication. Computer Supported Cooperative Work; 1996; Boston, Massachusetts, USA.
16. MacIntyre, B. and Mynatt E. D. Augmenting Intelligent Environments: Augmented Reality as an Interface to Intelligent Environments. AAAI 1998 Spring Symposium Series, Intelligent Environments Symposium; Stanford University, CA.
17. Mizell, D. Presentation given at VRAIS'98 Workshop on User-Interfaces for Wearable Computers.
18. Sabins, F. F. Jr. Remote Sensing Principles and Interpretation. San Francisco: W.H. Freeman and Company; 1978.
19. Sayers, C. P. and Paul, R. P. An Operator Interface for Teleprogramming Employing Synthetic Fixtures. Presence: Teleoperators and Virtual Environments. 1994 Fall; 3(4)
20. Siegel, J.; Kraut, R. E.; John, B. E., and Carley, K. M. An Empirical Study of Collaborative Wearable Computer Systems. Conference on Human factors in computing systems, CHI'95.
21. Sun Microsystems. <http://java.sun.com/beans/infobus/>.
22. Ullmer, B. and Ishii, H. The metaDESK: Models and Prototypes for Tangible User Interfaces. Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST'97), Oct. 14-17, 1997, Banff, Alta., CA.