Architectural Issues in Supporting Ad-hoc Collaboration with Wearable Computers

Gerd Kortuem, Steve Fickas, Zary Segall
Wearable Computing Research Group
Department of Computer and Information Science
University of Oregon, Eugene OR, USA
{kortuem,fickas,zs}@cs.uoregon.edu

ABSTRACT
This paper reports our experiences in developing *Proem*, an open software framework for building applications that support ad-hoc collaboration of wearable users. We discuss common tasks of such applications and identify a set of common services to be provided by a future collaboration framework. We propose a concrete collaboration architecture and discuss its properties.

INTRODUCTION
The use of communication technologies like phone, fax and email has become commonplace. Despite this fact most social interactions between individuals still occur when we meet people face-to-face, i.e. when we are in other peoples' immediate physical proximity, can talk to them, hear their voice, and when we have a chance to look into people's eyes or observe their gestures.

Many face-to-face conversations with other people are actually the result of chance encounters, i.e. situations where we meet someone unexpectedly, for example in a hallway or an elevator. A large majority of the people we encounter every day we do not know and have never met before, some however, we know personally and recognize. An encounter with another person is always a chance for striking up a conversation and for the exchange of information. Sometimes we use encounters to cooperate with other people and to pursue and advance our own goals. For example, we purchase items from a salesperson, exchange news and coordinate schedules with co-workers, or make travel arrangements with friends at home.

Chance encounters play an important part in our social life, but they are also vital for collaboration at our work place and for coordination of our work activities [1].

Wearable computing facilitates a new form of human-computer interaction comprising a small body-worn computer that is always on and always ready and accessible. In this respect, the new computational framework differs from that of hand held devices, laptop computers and personal digital assistants (PDAs). The same properties of wearable computers that enable a new form of human-computer interaction can also facilitate a new form of human-human interaction:

i. A wearable computer runs continuously, and is "always ready" to interact with its environment including other people and their wearable devices. In particular, this is true for situations where we unexpectedly encounter other people or where we are not aware of other people's presence.

ii. Wearable computers aware of their environment. Using sensors, a wearable computer is not only able to detect the presence (and possibly identity) of other people and their wearable computers, but it can also assess our communicative situation. For example, it can decide whether we are engaged in a private conversation with one individual or if we are in a public meeting with a large number of people.

iii. Finally, a wearable computer is proactive and uses its situational knowledge to assist and support the user without requiring user intervention. By using information about the people, we know and things we do, it is a highly personal device that is able to assist us in our interactions with other people.

All these properties make wearable computers well suited for supporting *ad-hoc collaboration in face-to-face situations*. It can augment our social interactions with other people and it can help us form and maintain cooperative and interdependent relationships with the people we meet. It can do that, for example, by identify people we want to meet, facilitate the exchange of information like business cards, and support us in more complex tasks like scheduling meetings, or trading delivery tasks [4].

EXPERIENCES DEVELOPING COLLABORATIVE WEARABLE APPLICATIONS
Over the course of the last year we have developed a number of collaborative applications that support wearable users in face-to-face settings. Initially conceived separately, we have begun the development of a common software platform that we call *Proem*[1] [3]. Our goal is to build an open collaboration framework on top of which applications for ad-hoc co-located collaboration can be implemented.

[1] A 'proem' is a brief introduction.

Proem devices are personal agents that act on our behalf and in our interest. Their main task is to communicate with other wearable devices as we encounter other humans, to exchange information with these devices and to engage in collaborative behavior.

Several simple applications were built on top of the Proem infrastructure [3,4]:

- An *awareness tool* that informs the user about the names and company affiliations of other *Proem* users in a meeting.
- A *reminder tool* that alerts users to the presence of specific people they want to meet or talk to.
- A *diary* application that keeps a record of all encounters during the course of a day.
- A *matchmaker* that alerts users to the presence of other people who match their interest profile.
- A *delivery task trader* that helps users to negotiate trades of delivery tasks with other users such that both users benefit.

While building these applications we identified a set of collaboration tasks that is common to all these applications.

Task 1: Identification of possible collaboration partners

The first task of any collaborative application is to identify possible collaboration partners. This task can be divided into three subtasks.

i. First, applications need to be able to tell which other wearable devices are in the immediate vicinity. This can be done using dedicated sensors or as part of a low-level wireless networking protocol like the one used in Bluetooth [5].

ii. Second, applications need to be able to determine which of these nearby devices are actually possible collaboration partner, i.e. which devices are able and ready to collaborate. In our approach, we assume that there is a large variety of heterogeneous wearable devices, which are not necessarily able to interoperate even if they are able to 'see' each other or to communicate. In particular, we do not assume that the same applications are installed on all *Proem* devices, but rather that individual users write their own applications. Thus, not all devices that encounter each other might be loaded with the same collaboration software. The question here is how an application can find matching applications on other devices.

iii. In case two or more users decide to engage in some collaborative activity, but discover they do not have compatible application software, there must be a way to download and install applications on demand. Applications can either be loaded from a centralized repository on the Internet or preferably passed along from one device to the next [2].

Task 2: Setting up and maintaining a collaboration session

Users are highly selective about their collaboration partners. While at any given moment there might be a large number of possible collaboration partners (for example at a conference or trade show), users typically only want to engage in collaborative activities with a small subset of these users. The task of setting up a collaboration session with a well-defined set of users entails the following subtasks:

i. First, applications must be able to tell other wearable devices apart based on the identity of their respective users. While Task 1 deals with device identities, setting up and maintaining a collaboration session is based on *user identities*. In a Bluethooth network devices that share the same area and that are within range of one another can potentially establish ad hoc connections between themselves. However, users want to have control over whom they collaborate with and thus applications must be able to tell to whom an individual wearable device belongs. Consequently, in order to take part in a collaboration session wearable computers must provide information about their owner's identities. This must be done in a secure manner which prevents spoofing and forgery.

ii. Once users have decided to collaborate with each other, i.e. the participants for a collaboration session are defined, applications must be able to set up a virtual private network over wireless links between the participants' machines. This network must guarantee security and privacy and must ensure that no one can listen in from the outside.

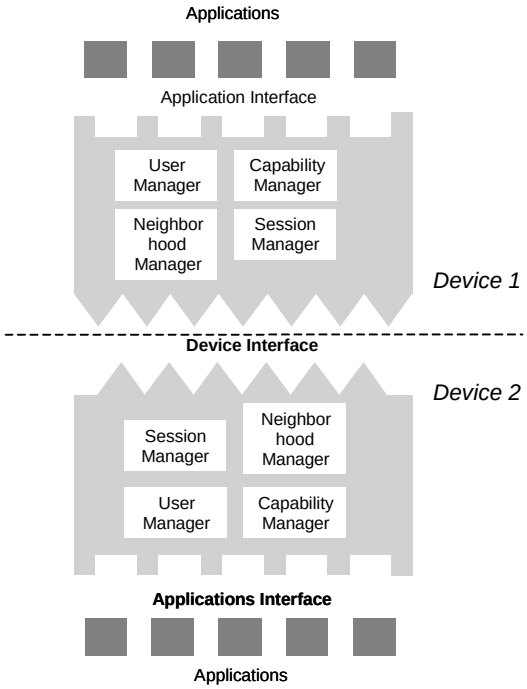
iii. While a collaboration session is going on, new participants must be able to join in at any time and other participants must be able to leave without compromising security and privacy.

Task 3: Reestablishing past collaboration sessions

Since in our daily life we repeatedly encounter or meet the same people, a collaborative application must be able to reestablish a collaboration session that occurred in the past. This involves three separate subtasks:

i. First, applications need to keep a history of collaboration sessions and they need to store state information with each of them: Who were the participants? Which actions were performed? What was the last session state?

ii. Second, applications need to be able to identify and recall a past collaboration session. Identifying a past session is not as trivial as it sounds. In general, collaboration sessions can be identified by the list of participants. So if the exact same group of users meets again, an application can easily identify which session should be restarted. However, what if only a subset of the original user group is present? Our a superset, i.e. new users want to join in? In such cases there might



be more than one session that potentially matches the current user group.

iii. Finally, applications must decide where to store state information. If available, state could be stored on a centralized server. Alternatively, state could be stored in a distributed and replicated fashion on all wearable devices. Both approaches have advantages and drawbacks: a centralized session storage requires a server that is accessible everywhere and thus would most likely require a wireless Internet or LAN connection. A decentralized approach is much more complex and requires that each wearable device carries complete state information for a potentially large number of collaboration sessions. On the upside, it guarantees that a session can be restarted solely with the data stored on wearable devices.

COLLABORATION ARCHITECTURE

It is obvious that writing applications for ad-hoc collaboration without proper infrastructure support is hard. There is a large set of basic functionality that must be implemented by applications. It thus makes sense to devise a software infrastructure or collaboration middleware that comprises a set of common services. Our *Proem* system is a first step in this direction, but only provides a subset of the required functionality [3]. *Proem* is based on a service architecture (which itself is based on Sun's Jini technology [6]) that enables the discovery of devices and their capabilities. Moreover, *Proem* provides the notion of device and user identities. That way it is possible to establish collaboration sessions with particular users and to prevent unauthorized users from joining. At this point, however, *Proem* does not implement security, so it is possible for users to forge identities.

Our goal is to add the missing functionality over time. We have thus devised a new and more comprehensive collaboration architecture that is depicted in Figure 1. The core of the architecture consists of four components:

- The *Session Manager* is responsible for maintaining collaboration sessions. It provides services for creating, joining, leaving, closing and restoring collaboration sessions.
- The *User Manager* maintains the user's identity.
- The *Capability Manager* maintains information about the device capabilities in terms of supported collaboration functions and installed applications.
- The *Neighborhood Manager* maintains up-to-date information about nearby devices and their capabilities.

In this architecture each device provides two interfaces:

- The *Application Interface* provides services required by the applications running on the device. This interface mainly exposes the functionality of the session manager.
- The *Device Interface* provides functionality to nearby wearable devices. This interface exposes the functionality of the capability manager, the user manager and part of the session manager.

CONCLUSIONS

This paper has identified requirements for a software framework for wearable ad-hoc collaboration. Through the experiences we gained while developing applications for *Proem* we were able to identify a set of common collaboration tasks related to collaboration sessions and user identities. Additional issues are related to privacy, security and on-demand installation of applications. We believe that the success in solving these issues and building collaborative applications will depend on the quality of the underlying architecture. We are currently specifying our proposed architecture in more detail and hope to evaluate it when we build new applications.

REFERENCES

1. Belotti, V. Bly, S. *Walking Away from the Desktop Computer: Distributed Collaboration and Mobility in a Product Design Team. Proceedings of CSCW '96*, ACM Press
2. Kortuem, G., Fickas, S., Segall, Z. *On-Demand Delivery of Software in Mobile Environments.* Nomadic Computing Workshop, 11th International Parallel Processing Symposium, April 1-5, 1997, Geneva, Switzerland.
3. Kortuem, G., Segall, Z., Cowan Thompson, Th.G. *Close Encounters: Supporting Mobile Collaboration through Interchange of User Profiles*. Proceedings 1st International Symposium on Handheld and Ubiquitous Computing (HUC 99), September 1999, Karlsruhe, Germany.
4. Kortuem, Gerd, Schneider, J.Suruda, J. Fickas, S., Segall, Z. *When Cyborgs Meet: Building Communities of Cooperating Wearable Agents*. Proc. 3rd International Symposium on Wearable Computers (ISWC 99), Oct. 1999, San Francisco.
5. Bluetooth Organization. www.bluetooth.org
6. Jini. www.javasoft.com/products/jini