Mobile Peer-to-Peer Technology used to Promote Spontaneous Collaboration

Alf Inge Wang, Carl-Fredrik Sørensen
NTNU, NO-7491 Trondheim Norway, alfw/carfrs @idi.ntnu.no

Thomas Fossum
NILU, NO2027 Kjeller, Norway, thomas@devbox.no

Abstract

Mobile devices have in the recent years been able to form peer-to-peer networks making it possible for people to share information and interact. This technology makes it possible to create new tools that initiate and improve human collaboration. The paper presents experiences from creating a prototype for spontaneous collaboration, supported through the appliance of peer-to-peer applications on wireless mobile devices with the ability to form ad-hoc wireless networks. The paper gives an example of an application that promotes human collaboration by using peer-to-peer technology, and describes the technological challenges faced when implementing such applications. The paper reflects on some user tests that were carried out using this prototype.

KEYWORDS: Spontaneous collaboration, peer-to-peer networks, mobile environments.

1 INTRODUCTION

Humans are said to be social animals. We seek contact with other humans, and we are used to collaborate to achieve our goals. By interacting in various ways, we can share information and exchange ideas and opinions. Both in professional and private settings, we use communication as a mean for collaboration. Collaboration is vital for productivity at workplaces. Many research projects have investigated how collaboration can be improved using computers.

Most of the earlier research studying human collaboration is based on the premise that interactions between colleagues happen in formal meetings where several persons have long interactions on pre-planned topics. Finn [9] states that the interaction patterns between people at workplaces are indeed different. Most interactions are short, and tend to be spontaneous. Such interactions can occur at any location at the workplace when two or more co-workers meet by chance. The discussions will usually be informal, and tend to be continuations of prior conversations. Informal, spontaneous interaction at work places has important functions both socially and for collaboration. Incidental encounters between co-workers are a natural setting for collaboration where a lot of important information, ideas, inputs and updates on various issues can be exchanged.

With hardware and bandwidth becoming increasingly cheaper, new devices, platforms and applications are constantly being developed. According to Ken Dulaney of Gartner Group [6], this can be expected to continue, and devices for computing and communication will become faster, smaller, more accessible, more affordable, and easier to use. This trend is already clearly visible through the fusion of PDAs and mobile phones.

The same level of innovation can be found within network technology. Wireless Local Area Networks (WLANs) have become common, and new standards such as Bluetooth and other implementations of Wireless Private Area Networks (WPAN) are being implemented in many new devices. Other areas of network technology are also subject to extensive research. Among these is ad-hoc networking, which enables devices to discover other devices autonomously and to communicate without the need for a central server or other existing infrastructure. This technology is very interesting for use in applications supporting spontaneous collaboration.

All these technologies and the new and improved ways of communication, provide new opportunities for collaboration. In this paper, we describe how mobile peer-to-peer technology can be used to promote spontaneous collaboration between co-workers. Our approach makes it easier to find the right person to collaborate with, and to share ideas and information utilising new technologies.

The rest of the paper is organised as follows: Section 2 describes two scenarios that were the motivation for building the tool. Section 3 describes the concepts of the tool. Section 4 presents the evaluation of potential technologies that could be used to develop the tool. Section 5 describes how the tool was developed, tested and experiences from building the tool. Section 6 describes related work, and finally Section 7 concludes the paper.

2 SPONTANEOUS COLLABORATIVE SCENARIOS

In this section, we will present two scenarios that promote spontaneous collaboration utilizing mobile peer-to-peer networks. The first scenario describes a university working environment with a large number of employees that are *knowledge workers*. The second scenario describes a large conference where a lot of unfamiliar people are gathered, and where interaction and collaboration are essential for gaining the maximum benefits of being present. The last part of this section presents the main characteristics and challenges of the two scenarios.

A day at the office at the University John and all his colleagues just recently installed an application called ProMoCoTo (Pro-active Mobile Collaboration Tool) on their mobile devices. The main objective of this application is to promote and support spontaneous collaboration. John is currently having some problems with Java RMI, and he adds this problem as keywords in a search criterion in his professional profile in ProMoCoTo. With ProMoCoTo installed, the mobile device now has the ability to match profiles with other users, and in this way initiate interaction between co-workers with matching queries and skills. The professional profile contains personal details and a detailed list of skills and current projects.

John walks through the corridor and his mobile device points out a person present with some experiences with Java RMI. He decides to talk to this person, a new PhD student, and the student suggests a solution to his problem. The use of ProMoCoTo has disclosed hidden knowledge at the university. The social barriers for initiating interaction with strangers have been reduced, since they now know that the other party can be a useful collaborative partner.

At the conference John travels to a large conference, with researchers from all over the world attending. The conference organisers have asked all attendants to install ProMoCoTo on their mobile devices in advance of the conference. When John arrives, he discovers that most of the others have done so. Since the conference has many parallel tracks, it is rather difficult for John to meet all the people working in his research areas. John configures his ProMoCoTo with the keywords "*distributed architecture and context-aware applications*". Thanks to ProMoCoTo, John meets several persons on his first day of the conference with the same research interests. ProMoCoTo also enables John to exchange contact information with all his new research friends. A particular useful feature of ProMoCoTo is the ability to show the picture of the owner of the device with matching profiles. This makes it a lot easier to pick the correct person in a crowd.

Characteristics of the scenarios

These scenarios are characterized by dispersed locations, where it is impossible to know experiences of all others. Further, both scenarios illustrate environments where people do not know each other and social barriers can prevent useful interaction. In such environments, a lot of "hidden" knowledge is unused because others do not know about it. Thus, a tool like ProMoCoTo will enable the workers to take the first step to utilise the hidden knowledge and to improve collaboration between colleagues.

Challenges found in the scenarios

The scenarios described above introduce some social and technological challenges that must be managed for the application to be useful.

The **social challenges** can be summarized in the three words: *Use, update* and *trust*. To efficiently share knowledge in an organization, most employees must *use the ProMoCoTo application*. If, e.g., large groups of the organization do not use the tool, their knowledge will still be hidden for others. In the same way, it is important for users to frequently *update their knowledge profiles* in order to reflect newly acquired knowledge. Lastly, it is important that the users *trust the application*, and that it is easy to choose what the users want to disclose to others.

There are also a number of **technical challenges** that must be managed to make the ProMoCoTo application useful in real environments. The application must be able to discover devices through the use of ad-hoc networks. Most wireless networks like infrared, WLAN, and BlueTooth have support for device discovery. Further, the application must be able to react to changes in the network to discover new peers or detect when peers are leaving. The application must also be able to perform search for matching peers to find the correct peers based on some specified search criteria. It is important that the information exchange between the mobile devices is very quick, since the ProMoCoTo application should be used in areas where people come and go. In some cases, this may cause that the information exchange will be interrupted and only parts of the information is exchanged. The application must be able to handle such situations. Lastly, it should be possible to run the application on various mobile devices, to increase its applicability and user base.

3 THE CONCEPT OF PRO-ACTIVE MOBILE COLLABORATION TOOL

The initial idea of the ProMoCoTo was to implement a mobile pro-active tool. This means that the tool can operate on its own without user intervention. The pro-active part of the tool includes discovery of possible partners for collaboration and the creation of ad-hoc networks where spontaneous collaboration can be conducted.

Since the tool should be used in a mobile environment, it should be possible to run the application on mobile devices. These mobile devices must support wireless networking that can work in peer-to-peer mode.

The main purpose of the ProMoCoTo is to promote collaboration that can occur when co-workers meet by chance, and engage in spontaneous, informal interaction and also information exchange. The application holds a profile containing owner's name, employer, current projects and other useful information. It must also include a detailed list of the owner's professional skills/interest. The user should set own preferences, where the search criterion is a vital part.

When two or more co-workers meet, their mobile devices will discover each other and create an ad-hoc network. This is done without any user intervention. The two or more devices in the ad-hoc network are now treated as peers in the peer-to-peer network. The various entities of ProMoCoTo will immediately start to exchange messages. Figure 1 shows how the messages are exchanged. First a query will be sent, and this query is run against the local profile of each peer. If a match is found, the profile will be sent to the remote peer who sent the query. When a profile has been received, the user is notified, and becomes aware of the possibility for interaction and collaboration with nearby users.

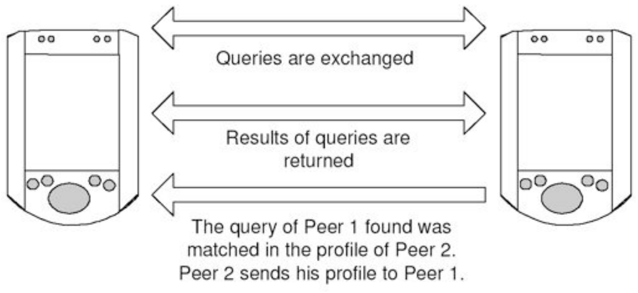

Queries are exchanged
Results of queries are returned
The query of Peer 1 found was matched in the profile of Peer 2. Peer 2 sends his profile to Peer 1.

Figure 1: Message exchange in ProMoCoTo

4 EVALUATION TECHNOLOGY

The initial criterion for selecting technology was that the tool should be possible to run on various mobile platforms. To enable portability, Java was chosen as the development and execution environment. In addition, the following requirements were used to determine suitability of existing peer-to-peer frameworks: It must be able to run on available resource-constrained devices, it must be able to support wireless ad-hoc communication, and it must provide autonomous peers, and the ad-hoc network must be truly decentralized. This means that the entire framework and tool must be able to run on the mobile device, without any need for external resources such as servers.

Evaluating Peer-to-Peer Frameworks

One of the main objectives of our peer-to-peer project was to evaluate existing peer-to-peer frameworks that can be used to implement peer-to-peer applications on mobile devices. The following candidates were evaluated:

JXTA [3] is a framework for developing P2P applications, originally initiated by Sun Microsystems. JXTA is now an open-source project supported by Sun. JXTA provides a set of protocols and APIs for general-purpose, computer-to-computer communication. The motivation behind the JXTA project is to create a common framework for P2P systems. JXTA provides a fully decentralized environment that enables user applications to connect to each other in an ad-hoc fashion. This enables application developers to create robust P2P applications with a minimum of effort. JXTA is implemented in a wide variety of programming languages, but the reference implementation is in Java. The framework is platform and network independent.

JXME [2] is JXTA for Java 2 Micro Edition (J2ME) and is a lightweight implementation of JXTA for mobile devices. It is specifically aimed at devices without sufficient computation and/or communication resources to participate in the network on their own. JXME is currently implemented in part on the CLDC/MIDP platform, and provides full JXTA functionality through the use of a relay host. The J2ME devices (mobile phones and PDAs) in the JXTA network are not connected directly to the JXTA network, and the relay hosts act on their behalf. All messages intended for the mobile devices are received and managed by the relay host. This means that the mobile devices do not communicate through direct peer-to-peer networks. There is also a JXME proxiless initiative, but no working implementation exists.

Proem [12] is a framework for developing and deploying P2P collaborative applications in a mobile ad-hoc networking environment. The framework is developed at the Wearable Computing Laboratory at the University of Oregon. The main objective of Proem is to provide a common framework for rapid development of applications for ad-hoc network environments. The framework is implemented in Java, and can be run on various wireless mobile devices. Proem is designed to be independent of underlying network transport protocols, and can be implemented on top of TCP/IP, HTTP, Bluetooth and others. The P2P applications that run on top of the Proem framework are called *Peerlets*. These are hosted in the Peerlet engine, which is responsible for instantiation, execution and termination of Peerlets. The Peerlet engine also controls the discovery of peers and users, the communication between Peerlets, and the Peerlet user interfaces. The Peerlet engine must run on every peer in the Proem network. Peerlets react to and communicate via events. The Peerlet engine fires events to Peerlets as a reaction to changes in its internal state or as a reaction to messages received by remote peers. Peerlets handle events asynchronously.

RockyRoad [1], has a set of features similar to JXTA. The main difference is that RockyRoad can be deployed on wireless devices with Palm OS and Pocket PC operating systems. However, this framework only provides limited support for mobile devices.

The peers involved in our prototype had to be truly autonomous. JXME is intended for mobile devices without sufficient computation and/or communication resources to participate in the network on their own. Also, previous experience with this framework concluded that the framework was not yet mature, and was missing important features in the API.

JXTA and Proem are similar in their overall approach, but their goals are different. Both JXTA and Proem are P2P frameworks as opposed to specific applications. JXTA has a broader scope and is more generic; it is intended as a common communication infrastructure for P2P applications, covering a large number of hardware platforms, networks, and applications. The Proem framework has a more narrow focus on ad-hoc networks and collaborative networks. Both frameworks are protocol-based; they define application-level communication protocols. There are also some fundamental differences between the two frameworks. The JXTA framework has its strength in support for developing P2P applications for semi-stable environments like the Internet. Proem, on the other hand, focuses on the person-to-person collaboration aspect of communication, and is designed for highly dynamic environments, such as those of ad-hoc networks with their constantly changing topology and available resources. Since JXTA is more general, this framework is more low-level than Proem, and requires more time to develop P2P applications.

All in all, we found Proem to be the most suitable P2P framework for our prototype. As the Proem framework was chosen for developing the prototype, most of the other choices of technology were given. The original version of the Proem framework runs on Java2 Standard Edition (J2SE), but a mobile version provided can run on PersonalJava or JDK 1.1.8. Our choice of framework also meant that we could only use mobile devices that have virtual machines that support the Java versions mentioned above. Our choice of framework made it impossible to use mobile phones and PDAs running the operating systems Symbian or Palm that only support Java2 Micro Edition (J2ME). However, for Compaq iPAQ we found the Jeode JVM, which is a PersonalJava distribution that could run Proem. Another option was to install Linux on a Compaq iPAQ or use a Linux-based PDA that could run a minimal version of the JDK 1.1.8 version.

5 THE ProMoCoTo PROTOTYPE

This section describes how the prototype was developed, tested and some experiences we gained from building it.

Developing the Peer-to-Peer Application

The ProMoCoTo tool was implemented as a Peerlet in the Proem architecture as shown in Figure 2. The prototype was implemented in one package to avoid problems with the Jeode JVM's inability to handle complex package hierarchies. Our prototype uses the underlying layers in Proem to handle discovery of nearby devices, message passing, identifying users and groups, event handling, and debugging. The Proem framework also provides support for GUI, but this feature was not used because of its limited functionality.

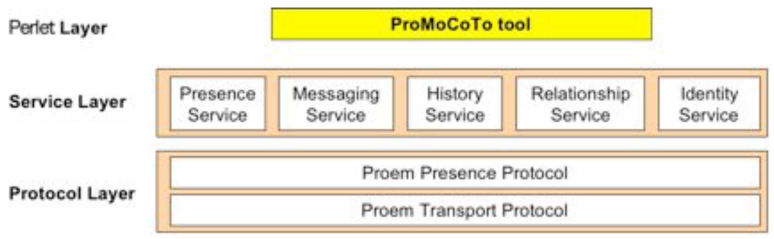
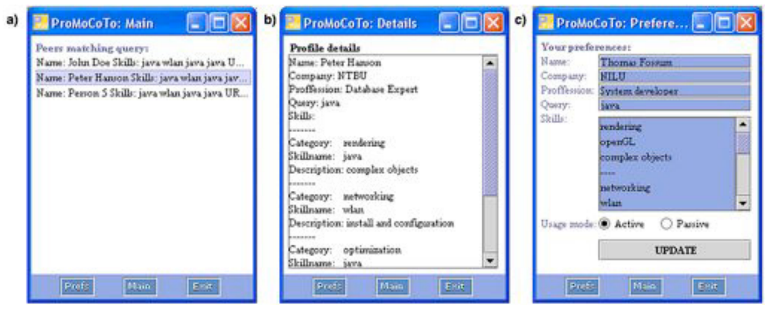


Figure 2: The ProMoCoTo tool shown in the Proem architecture

To handle the communication between peers, we implemented five message types: **PROMO_QUERY** - All peers send a query when they encounter another peer, **PROMO_CONFIRMATION** - The remote peer sends *CONFIRMATION* if the query returns a result, **PROMO_REJECT** - The remote peer sends a *REJECT* if the query does not return a result, **PROMO_REQUEST_PROFILE** - If a confirmation is received, the remote peer's profile will be requested, and **PROMO_SEND_PROFILE** -This message contains the profile of the local peer.

Figure 3 shows the three screenshots of the user interface of the prototype. Screenshot **a** shows the GUI for displaying users that match the query of the current user. Screenshot **b** shows the profile details of one of the persons found that match the query. Screenshot **c** shows the preferences screen where the user can specify personal information (name, company, profession, skills etc), entering queries to search for matching users, and choosing between passive and active usage mode. In the *passive* mode, ProMoCoTo just collect information about other users that match the query without notifying the user. In *active* mode, ProMoCoTo will notify the user if there is another user nearby that matches his query.



Peers matching query:
Name: John Doe Skills: java wlan java java U...
Name: Peter Hanson Skills: java wlan java jav...
Name: Person 5 Skills: java wlan java java UR...

Profile details
Name: Peter Hanson
Company: NTBU
Proffession: Database Expert
Query: java
Skills:

Category: rendering
Skillname: java
Description: complex objects

Category: networking
Skillname: wlan
Description: install and configuration

Category: optimization
Skillname: java

Your preferences:
Name: Thomas Fossum
Company: NILU
Proffession: System developer
Query: java
Skills:
rendering
openGL
complex objects
----
networking
wlan
Usage mode: [✓ Active] Passive

Figure 3: Screenshots from matching in ProMoCoTo

Testing the Peer-to-Peer Application

This section describes experiences from running the peer-to-peer application. The test was based on the university scenario described in Section 2.

The tests were performed in the hallways of Department of Computer and Information Science and the Norwegian University of Science and Technology (NTNU). The test persons were employees and students of the department since they were the target users for the application developed (see Section 2). This meant that all participants in the tests were computer experts and able to provide useful feedback. In total 7 test persons were used, but not all of them were used at the same time. We helped the participants to enter their profile in the ProMoCoTo, and then asked them to solve a particular programming problem in a project. This programming problem was entered as a query in the tool. To make this test interesting, we picked the programming problems to fit the skills of the participants. During the tests, the authors of this paper observed the involved participants and they were finally questioned about their experiences. Figure 4 shows two participants running the tool (left), and the tool running on an iPAQ (right). In this test we wanted to look at user and implementation issues as described in the paragraphs below.

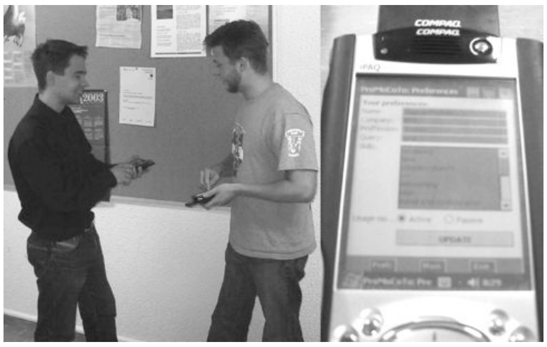


Figure 4: Testing ProMoCoTo peer-to-peer application

The first aspect we wanted to investigate was if the user experiences of using such a tool would promote spontaneous collaboration. We found that the participants with matching queries and profile found each other when they passed in the hallway, and they started to talk about the programming problem when the tool had notified the match. Initially, it was rather cumbersome to match participants as there was no sound notification when matches where discovered. This caused the various participants to walk around staring at the PDA screen to see if there were any matching persons around. In a later version of the application, a beep was introduced when a match was found. The users found the possibility to view the profile of other users that matched the query useful. This information was used to initiate the conversation and made it easier to discuss the problem (as they knew each others background). The participants said that the prototype would be even more useful if it could provide other collaborative tasks like file exchange. The prototype was judged to be easy to use and understand because of the limited functionality. Thus, if the ProMoCoTo should be extended, the more advanced functionality should be hidden in separate screens to avoid confusion. The main problem we discovered was to identify the correct user when several users were located in the same room. A picture in the profile could have solved this problem. Another approach could be to use unique sounds to identify matching devices, but this could disturb other people.

The prototype was deployed on Compaq iPAQ PDAs, which at that time were among the most powerful PDAs. The Compaq iPAQs were equipped with WLAN cards used for peer-to-peer communication. We found that this PDA was powerful enough for running the Java-based peer-to-peer application. The application was running fast enough to handle profile search of several nearby devices and providing the user with responsive user experience. One minor problem with the Pocket PC OS on the iPAQs was that it was hard to get the debug information since there was no easy way to redirect the standard output stream when running Java applications.

We tested the network performance to see if it was fast enough for two or more devices to exchange the necessary data. The time for data exchange between to devices can be decomposed into the time to perform the discovery of nearby devices, to create an ad-hoc network between the devices and to exchange the data. Within the office building of the department, the range of the wireless cards was about 20 meters. The network performance was acceptable for typical situations where people are passing each other in a normal walking pace to create an ad-hoc network, search for profile, and exchange information before user were out of reach. The average time for performing the necessary network communication between two iPAQs running ProMoCoTo was about 10 seconds. We also tested the number of users that could be handled without any problems. We found that the prototype worked well with five simultaneous users, but more could cause performance problems. We also found that we could improve the performance drastically by sending the profile directly instead of discovering nearby users first.

Before we got the prototype to run correctly, we had to solve some initial problems. The mobile device would not have an assigned IP address if the wireless card were not in contact with a network. This meant that if the PDA were powered on in an area without any wireless access points or other peers, the prototype would not work. With no IP address assigned at start-up, the application would throw an exception because it was unable to join a MultiCastGroup. In order to prevent this problem, IP-addresses were set to 10.0.0.1, 10.0.0.2, 10.0.0.3, etc., on the participating devices to get consistent networking performance. The wireless cards had to be inserted before application start-up.

We also tested portability of the prototype by deploying the tool on two different platforms, namely the Pocket PC OS using the Jeode JVM (running on the iPAQ), and the Windows XP OS using the JDK 1.1.8 JVM (on a Acer Travelmate laptop). Both versions were compiled with JDK 1.1.8, and the class-files were deployed on the iPAQs. Some modifications in the code were necessary to make the prototype run on the iPAQ. These modifications were all related to the fact that relative paths to files and images were not handled well on this platform. To conclude, our prototype could run on different mobile devices by making small changes in the code.

Experiences

The actual implementation of the prototype was a time-consuming effort, due to an unfamiliar framework and initially unsolved problems. The poor documentation and lack of relevant sources for information on development using this framework made the early stages of the development a somewhat frustrating experience. But important lessons were learnt by trial-and-error. The choice of using an existing P2P framework for development instead of building everything from scratch proved in the end to be a sensible choice. The proof-of-concept could not have been realized without the use of the Proem P2P framework. Well into the development process, we realized that more investigation of the built-in profiles in Proem could have provided an alternative solution to what we were implementing. Still, the implementation proved its worth through the experiences gained and the test results it provided.

The Proem framework proved to be a sensible choice although the documentation was not too extensive. The framework offered all the functionality the prototype needed, and it ran on the iPAQs without too much hassle. It was in reality the only real choice, as other frameworks had a somewhat different scope and focus. The team behind Proem was very supportive, and was of great help during the development process.

6 RELATED WORK

In [10, 8], the concept of IPADs is presented. IPAD is used as a collective term for mobile, portable devices intended to improve the opportunities for face-to-face interaction in a group of people. IPADs aim to provide awareness solutions that combine the advantages of desktop-based awareness applications; constant, non-disruptive awareness, with the mobility provided by mobile devices. The IPAD is used as a contact facilitator, initiating contact and extending the human range of awareness. Several experimental prototypes have implemented this concept.

The Hummingbird [11] has its name because it ”hums” whenever another hummingbird is nearby. The prototype implements the concept of an IPAD, and is meant to promote presence awareness and collaboration between people who are in the physical vicinity of each other. Constant awareness information is supplied to the users that are close to each other, without any reliance on an underlying infrastructure. When another device has been detected, a sound is generated, and the name of the owner of the other device is displayed on the device’s screen. Studies show that the Hummingbird did increase awareness between group members, and that it had the potential to complement other forms of communication such as phone and email. The Hummingbird prototype was found to be particularly useful when a group of people were in an unfamiliar location, with no other form of communication available.

The main objective of Proxy Lady [4, 13] is to promote informal, opportunistic face-to-face communication. Proxy Lady is intended to be used by people who meet frequently, such as co-workers, and needs to share information when opportunities occur. The system supports these informal interactions by providing location awareness, as well as information items such as email messages and other files. These items are stored on the mobile device during synchronization with a desktop computer. The concept behind Proxy Lady assumes that these information items can be used as the basis for informal interaction. Users may want to discuss the information items they carry. This implies that the users of the system already know each other, and store information items on their mobile devices with later interaction in mind. As the Hummingbird, the Proxy Lady prototype provides presence awareness [5], but it adds the possibility to filter the candidates for interaction by what information items they carry.

AIDA [14] is a prototype that combines the features of an Instant Messaging application with those of a Presence Awareness application. The prototype is deployed on a mobile device and is used to interact with devices that come within reach of the user as he moves within a pervasive computing environment. AIDA uses an infrastructure called DoMo, for opportunistic interaction with services available in a pervasive computing environment. When users join the ad-hoc network, agents representing services are registered in a list held on each mobile device. The user has access to a given set of services offered by other users of this network. The application also provides presence awareness through an Instant Messaging-part of the application.

The prototype Hocman [7] is a mobile HTTP P2P application, which supports social interaction and enabled sharing of multimedia content over an ad-hoc network. The prototype has been deployed on mobile devices and tested in the very mobile environment, namely motor bikers riding their bikes. A rapid peer discovery algorithm was invented in order to cope with the very brief opportunities for peer discovery and the creation of ad-hoc networks.

The concept and prototype described in the paper was inspired by several of the concepts described above. The aspects of awareness and the initiation of interaction through the use of mobile devices are central. The developed prototype share many similarities with the concept of IPADs. It is intended for raising awareness of co-worker's skills, and to initiate interaction that can lead to useful collaboration. Also, the prototype is independent of existing network infrastructure, and only need the presence of other devices to be truly useful.

Our prototype shares the feature of providing information on nearby peers with implemented concepts such as the Hummingbird and the Hocman. This aspect is also implemented by the AIDA prototype. Both the Proxy Lady and the Hocman prototype add the concept of exchanging information items. This can however be done without additional software once the devices have established an ad-hoc network, but it could nevertheless be a useful extension of our prototype, providing increased convenience for the users. The filtering of nearby users based on what information items they carry, used in both the Proxy Lady and Proem, closely resembles the concept that will be used in the future prototype to determine what peers that are interesting for collaboration.

Our prototype combines the strengths of several of the described prototypes. In addition to the presence awareness, our prototype uses the concept of filtering interesting remote devices, through matching search criteria to fire a notification and further exchange of information. The information on the owner of the nearby device is accessible on the local device through the exchange of profiles. The main technical difference with our approach is that our prototype can run on general-purpose mobile devices and that it uses Java.

7 CONCLUSION

In this paper we have presented some experiences from implementing a collaborative application using peer-to-peer technology in Java. We have found a promising application area for peer-to-peer technology that can promote spontaneous collaboration and improve knowledge sharing in big organizations. We have also found that peer-to-peer frameworks for Java are still immature, which makes it difficult to make stable and useful peer-to-peer application in Java. In addition, the lack of peer-to-peer frameworks in J2ME causes Java-based peer-to-peer applications to be limited to only the most powerful PDAs. Such applications have a much broader audience if they can be run on mobile phones. The authors of this paper are involved in a project developing a new framework for developing cooperative applications in J2ME called Peer2Me (see www.peer2me.org). We have successfully tested an early version of this framework on mobile phones (Nokia

6600 and Sony Ericsson P900) using Bluetooth. The framework provides opportunities to create a palette of collaborative applications for mobile devices utilizing peer-to-peer communication.

Acknowledgement
This paper is a result of work in the project MOWAHS sponsored by the Norwegian Research Council's IKT2010 program.

References

[1] Offshore Algorithms. Rockyroad/jrra. Web: http://www.jrra.org, 2004.

[2] C. W. Akhil Arora and K. S. Pabla. jxme: JXTA Platform Project. Web: http: www.jxme.org, February 2005.

[3] J. Brendon and J. Wilson. *JXTA*. New Riders Publishing, 2002.

[4] P. Dahlberg, F. Ljungberg, and J. Sanneblad. Proxy Lady: Mobile Support for Opportunistic Interaction. *Scandinavian Journal of Information Systems*, 14:3–17, 2002.

[5] Per Dahlberg and Johan Sanneblad. The use of Bluetooth enabled PDAs. In *Proceedings of IRIS-23*, Uddevalla, Sweden, August 12-15 2000.

[6] Ken Dulaney. The Wireless and Mobile Market Starts to Mature. In Monica Basso, editor, *PREDICTS 2003, Gartner's Predictions for the year ahead - Wireless & Mobile*. 22 November 2002.

[7] Mattias Esbjörnsson, Oskar Juhlin, and Mattias Östergren. The Hocman Prototype - Fast Motor Bikers and Ad Hoc Networking. In *1st International Conference on Mobile and Ubiquitous Multimedia (MUM 2002)*, Oulu, Finland, December 11-13 2002.

[8] Roberto Silveira Silva Filho. Awareness and Privacy in Mobile Wearable Computers. IPADS: Interpersonal Awareness Devices. Final report for ICS234A - Virtual Colocation, Web: http://www1.ics.uci.edu/ rsilvafi/papers/VirtualColocationFinalPaper.pdf, 2004.

[9] Kathleen E. Finn. *Video-Mediated Communication*. Lawrence Erbaum Associates, Inc., April 1 1997.

[10] L. Holmquist, J. Falk, and J. Wigström. Supporting group collaboration with inter-personal awareness devices. *Journal of Personal Technologies*, 3(1–2):13–21, 1999.

[11] Lars Erik Holmquist, Joakim Wigstrom, and Jennica Falk. The Hummingbird: Mobile Support for Group Awareness. In *Demonstration at ACM 1998 Conference on Computer Supported Cooperative Work*, 1998.

[12] Gerd Kortuem. *A methodology and software platform for building wearable communities*. PhD thesis, University of Oregon, December 2002.

[13] Fredrik Ljungberg, Per Dahlberg, and Johan Sanneblad. Supporting opportunistic communication in mobile setting. In *ACM 2000 Conference on Human Factors in Computing Systems (CHI 2000)*, pages 111–112, The Hague, The Netherlands, April 1–6 2000.

[14] Christian Navarro, Jesus Favela, and Marcela Rodriguez. Extending instant messaging to support spontaneous interactions in ad-hoc networks. In *Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments*, New Orleans, Louisiana, USA, November 16–20 2002.