MP-Collaborator: A Mobile Collaboration Tool in Pervasive Environment

Bruno M. C. Silva[1], Paulo A. C. S. Neves[1,2,3], Mieso K. Denko[4], and Joel J. P. C. Rodrigues[1,2]
[1]Department of Informatics, University of Beira Interior, Covilhã, Portugal
[2]Instituto de Telecomunicações, NetGNA Group, Covilhã, Portugal
[3]Polytechnic Institute of Castelo Branco, Castelo Branco, Portugal
[4]Department of Computing and Information Science, University of Guelph, Guelph, ON, N1G 2W1, Canada
bruno.silva@it.ubi.pt, pneves@co.it.pt, denko@cis.uoguelph.ca, joel@ubi.pt

***Abstract*— In modern organizations, the communication between collaborators is essential to improve productivity. There is a need for mobile collaboration tools that allow efficient collaboration among staff in organization which may be located in different geographical areas and time zones. Mobile and pervasive computing provides easy and convenient access to information, enabling effective collaboration. This paper describes a context and location-aware mobile application, called MP-Collaborator, created to improve and optimize the communication between collaborators in any organization. MP-Collaborator draws information from location, user status and Pocket Outlook Calendar to create a user availability profile, which is published to any other user in the network. Based on availability status, the user can be reached using a simple phone call or a message. Through the opportunistic use of available mobile technologies MP-Collaborator provides a simple but critical service - user presence service. The proposed solution is validated both in terms of features and communication through a series of experiments on real devices through Wi-Fi network.**

***Index Terms*— Mobile computing, location-aware computing, mobile collaboration tools, peer-to-peer communication**

I. Introduction

Many organizations rely on collaboration tools to increase productivity, communication between working teams and personnel, thus effectively managing staff potential [1]. Teams can sometimes reach hundreds of professionals distributed over several countries [2].

In the search for improving collaborative work over the Internet scientists at CERN developed the hypertext markup language (HTML) and the world wide web (WWW) [3]. The Internet became a pervasive technology for collaboration and an absolutely indispensable tool for years to come.

The advent of mobile communications supported on smart mobile devices like smart phones and personal digital assistants (PDAs) offer a unique opportunity to develop efficient tools for mobile collaboration while on the move. Moreover, these devices present features that only fixed computers could offer not so long ago, such as email and Internet connectivity. Mobile devices can accelerate work and simplify coordination processes [4].

Companies are spending fortunes on communication between personnel for collaborative work. However, sometimes a decision is not taken because a person is not available, does not answer the phone or, basically, is unreachable. More than knowing the location of such person, it is important to acknowledge the person's current status, thus creating a user profile. The user profile must provide enough information about availability, location and possible communication means. Then, location-awareness must be combined with context-awareness, in order to effectively allow communication between the different human resources of a given corporation, managing user's presence inside the corporation.

The popularity of instant messaging applications can be explained easily with the combination of real-time chat and profile status exchange. However, such approach implies that the other user is also connected to reply. When a mobile scenario is taken into account, constant data networks connectivity greatly impacts battery life and communications costs, while GSM/UMTS may be available, allowing a simple phone call or short message (SMS). Our application takes advantage of current mobile technologies in behalf of the user, where other approaches lack [5-7].

The MP-Collaborator application is all about user presence inside an organization. This mobile application augments a mobile device by combining location information, mobile communication (call and SMS), scheduling information (through Pocket Outlook Calendar), creating and sharing the user profile. The communication paradigm allows server-less user interaction, through peer-to-peer communication among different mobile nodes.

Mobile ubiquitous computing has evolved greatly since the appearance of the adequate mobile devices such as cell phones or PDAs, which feature technologies and functionalities such as Wi-Fi, 3G/GSM, GPS receiver; with processing power and

978-0-7695-3841-9/09 $26.00 © 2009 IEEE
DOI 10.1109/WiMob.2009.65

IEEE computer society

storage capabilities that overcome desktop computers of the late 90's. In this context, MP-Collaborator is a useful mobile application that fits into the everyday needs of any company or organization in the modern world. Furthermore, to the best of our knowledge, it is a new and innovative mobile application for organizational environments.

MP-Collaborator presents many advantages over existing approaches like the use of different communication capabilities (SMS, phone call), the diffusion of user profile based on location and schedule, while also preventing inadequate communication attempts, together with the peer-to-peer nature of communication.

The rest of the paper is organized as follows. Section II Presents the related work with focus on mobile and pervasive computing, location-based services, and other similar approaches. Section III describes application development including its requirements, architecture and the technologies employed. Section IV evaluates our approach, presenting user interface and application functionality along with experimental evaluation and validation. Finally, section V concludes the paper and presents some possible future work.

II. Related Work

This section briefly reviews related work about mobile and pervasive computing, context and location aware services, location-based services, and similar approaches.

A. Mobile and Pervasive Computing

Marc Weiser [8, 9] presented in the early nineties his vision of ubiquitous computing, presenting the concept of a complete user abstraction of the technology where the machine interacts with man. However, Weiser's vision was lacking technological support that we have now [9]. Internet is an example of a pervasive technology, where users are interested on the information and services and not the technology itself. A user accesses the Internet without knowing any configuration, network architectures or protocols, on almost any device in many places around the globe [10]. Ubiquitous computing overcomes the device, the network, and the protocol. Technology will interact with us in the same way a friend help us, technology will act on behalf of the user in a transparent way, and not by user request.

The society is changing and reorganizing its behaviors, devising new ways of communication, collaborations and organization. These new ways of action will provoke changes so profound that in some cases, they result in complete new social behaviors [11].

With the appearance of mobile devices featuring wireless communication, computing evolved to a new level, taking advantage of portability, mobility, and everywhere connectivity. However, mobile computing still has some technology barriers to overcome, such as the quality variation of a wireless network, local access limitation, or even energy limitations. These problems in mobile computing have a directly effect in the computing experience [11].

Ubiquitous computing takes advantage of mobile computing, adding several parameters as Figure 1 presents. Spaces must be increasingly smart with sensor technology to capture relevant data on behalf of the user, while invisibility is one of the major characteristics that a ubiquitous system must have. Uneven scalability is a major issue since not every location will have the same capabilities, and the system must take advantage of the available resources, and not cease operation due to the lack of a particular resource.

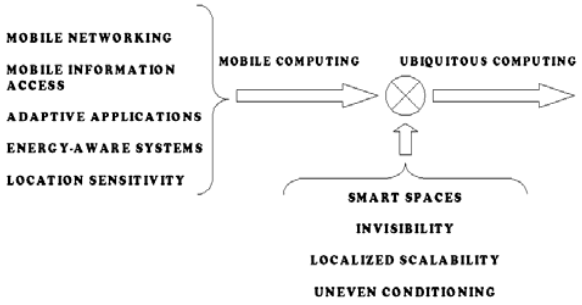


Fig. 1. Relationship between mobile and ubiquitous computing [10].

A major example is through disconnected operation, where the system must operate even in the absence of communication [11].

B. Context and Location Awareness

A truly pervasive system must be sensitive to context and location. A pervasive system needs to be able to create a user profile based on collected knowledge of, for instance, a user's actions, location, health condition, and daily behaviors. This knowledge will help the system to assist the user in daily actions, inclusively taking decisions on its behalf. That is the essence of a true pervasive system.

However sometimes it is not an easy task to combine many different and sometimes contradictory sources of information, such as the current user location and user appointments. Maybe the user should be in a meeting, but he is currently having lunch, or the lunch is itself the meeting!

Context and location awareness brings an important issue sometimes overlooked on ubiquitous computing: the privacy issue. Users generally do not feel comfortable with constant monitoring of their daily activities.

C. Location Based Services

The Location-Based Service (LBS), installed in mobile devices, is assuming an important role in today mobile applications. Examples of LBS systems come from traffic routes control and vehicle tracking, where commercial applications already exist.

One of the first LBS appeared in 1996, in the USA, when the senate approved the mandate Enhanced 911 (E911). The E911 gave permission to mobile network operators to track the location of a cell phone using GPS, and give it to the proper authorities [12]. The E911 turned out to be a disappointing investment, and many operators tried to overcome the economic failure by launching a huge number of LBS. However they came to be an economical disaster, mainly due to the limited precision that the location service provided. The success of the LBS market as we know today began in 2005 with the appearance of evolved position technology and tracking systems with enhanced precision, namely Assisted GPS (A-GPS) technology, which transformed the LBS market in a growing success.

Other LBS beside GPS, gained also their position in the market. An example is services based on Radio Frequency Identification (RFID) technology, which uses radio waves to identify objects. A good example of a successful RFID-based system is the digital passport, already in use in some countries. The passport contains a microchip attached to a RFID antenna, allowing wireless capture of information by a receiver [13]. Another example of LBS technology is the Wi-Fi positioning technology, which is implemented in many devices, such as Symbian S60 9.3 OS devices. For instance on [14] the authors present a novel two-phase approach to Wi-Fi location sensing using pattern-matching for general location and logistic regression for finer-grained location.

D. *Existing similar approaches*

A collaboration tool for learning groups, featuring PDA and phone support through WAP is presented in [7]. The portal-based tool is implemented in Roxen web and application server with an integrated MySQL database. Student and lecturers communicate through a *news* server, *chat* server and Simple Mail Transfer Protocol (SMTP) server. An external provider also allows Short Message Service (SMS) to push information.

The *TouchSync* framework allows creation of user groups that can share data on their devices [5]. The framework can transform certain single-user applications to enable collaboration. The approach was validated with PalmOS and Bluetooth technology and a set of three applications: Calendar, Memo and Doodle.

The learning environment, sharing contents universally through WAP technology, motivates collaboration in the first example. Communication between participants is achieved by using other programs such as newsreaders, chat clients or e-mail. As a result the solution is not an “all-in-one” solution, forcing the user to switch applications. On the second scenario the solution is based on intrinsic technologies of a now deceased platform, PalmOS.

The user can benefit from the familiar interface of native applications in *TouchSync*. However collaboration is highly limited by the range of Bluetooth communications.

Although not created as a collaboration tool, InHand [6] is a mobile application designed to assist a mobile salesman, featuring context and location aware features. InHand also uses GPS to track user location, but needs a central server with a database to operate. Moreover, each user works unaware of each other, so application goal is different, although technologies can be similar.

III. Application Development

In this section, we will discuss the requirements of the MP-collaborator tool, the application architecture and the technologies required for its deployment. Our emphasis will be on existing as well as emerging architectures that foster its practical deployment.

A. *Requirements*

A mobile application has specific requirements that MP-Collaborator must also cope with. The user interface must be as easy to use as possible, with minimal input from the user. Screen size and orientation; together with stylus input minimization dictate an interface with large buttons, softkey use and appropriate font size, to enable on-the-move application use.

MP-Collaborator is not just another mobile application. It must offer services constantly; deal with communication disruption, users coming online and going offline, immediate profile creation and availability.

The user profile must aggregate the user location, schedule (obtained through Pocket Outlook Calendar), and information that the user itself may give. As a result user status uses current profile in order to define if a given user can be reached or not. Moreover, the application must use currently available communication tools; such as cellular network through SMS and voice call features.

The peer-to-peer operation of MP-Collaborator complicates the design, mainly due to the client/server model nature of current mobile applications. As a result, application operation must be radically changed, since no central server exists. Figure 2 presents the operational view of the application. The information data is routed across the devices that subscribe data. Location information can be obtained through several technologies like GPS (in use) and RFID. Pocket Outlook also helps with location information when available, buit is used mainly for the user profile, identifying current user activities.

One of the most important requirements of MP-Collaborator application is the continuous monitoring of all the context and location-aware information from other users. This feature will allow a user to know almost in real time the location and context of another user (profile), thus providing user presence information.

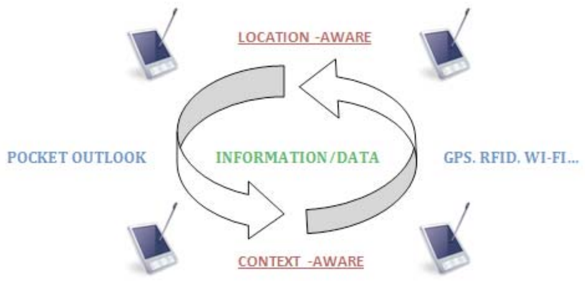


Fig. 2. Mobile application operational viewpoint.

Privacy concerns can also be addressed on MP-Collaborator, namely social disclosure of location and possible personal schedule sharing. However on the current version we did not took into account such issues, reserving them for future work.

B. Application architecture

The application architecture and its working principle are centered on a wireless network peer-to-peer technology. Figure 3 shows a scenario consisting of peer-to-peer devices, location databases and associated services in a wireless networking environment.

Each MP-Collaborator device connects to a wireless network (Wi-Fi at this stage), transferring and receiving the context and location information of all the connected and identified users. The context and location data is provided by Pocket Outlook Calendar application and by the location based service (LBS) in use (GPS, Wi-Fi position technology, or RFID).

The application database contains all the users' personal information and all the location information. After receiving the context and location data, the application will compare it to the database data, establishing user's profile.

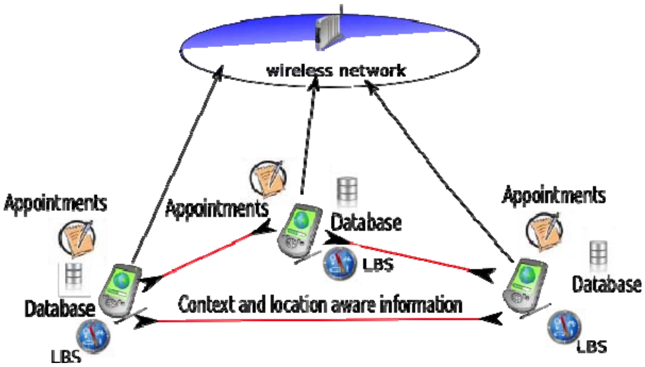


Fig. 3. System Architecture.

MP-Collaborator begins to operate initially in standalone mode. It searches for other connected devices, and if none is found, it assumes it is the current server, serving current user's information. Another MP-Collaborator device joins the network and searches for connected servers. Since a server is present, it connects to it, while also serving its own user's data. As a result both devices are connected and playing both client and server roles.

In a "steady-state" several MP-Collaborator devices are present and users' profiles are distributed among them. Several client/server connections take place simultaneously on the network.

C. Technologies

The mobile application targets Pocket PCs running Windows Mobile 6 and 6.1 (wm6/6.1). Two major solutions for application development exist: .NET Compact Framework and J2ME. The decision went for the Microsoft solution, mainly due to the following two features: availability of version 2.0 on every WM6.0 device without need for further installations/updates on the device, and tight interaction with a suitable database engine – SQL Server CE. Moreover, Java virtual machines (JVMs) on windows mobile are typically equipment manufacturer dependant, sometimes even in the same manufacturer different VMs exist.

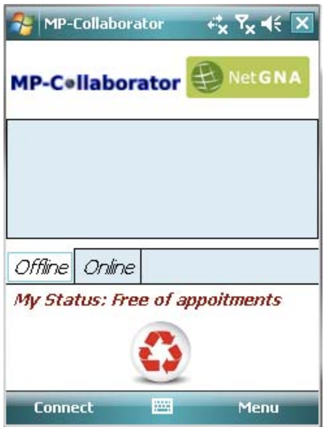
The Microsoft Visual Studio 2008 PRO IDE (Integrated Development Environment) offers the Microsoft Windows SDK V5.0, so we upgraded it to the V6.0. Both versions offer emulation debug for easy initial debugging of mobile applications. The IDE also offers a local database management tool.

The mobile application was developed in C# [15], mainly for the sake of existing know-how of the authors. The communication between devices uses the Transmission Control Protocol (TCP) over Internet Protocol (TCP/IP). To create a server that waits for connections we use the C# class TcpListener, and to create a client we use the C# class TcpClient, Both classes are included in the System.net.Socket namespace.

IV. Evaluation and Validation of MP-Collaborator

A. User experience and the MP-collaborator interface

This section presents an overview of the mobile application and its use in practical deployment. The MP-Collaborator presents a simple interface, which is user-friendly. The primary window is very intuitive, presenting the *manual refresh* button, the *connect* button and the application Menu. Figure 4 presents the main application interface.



My Status: Free of appoitments

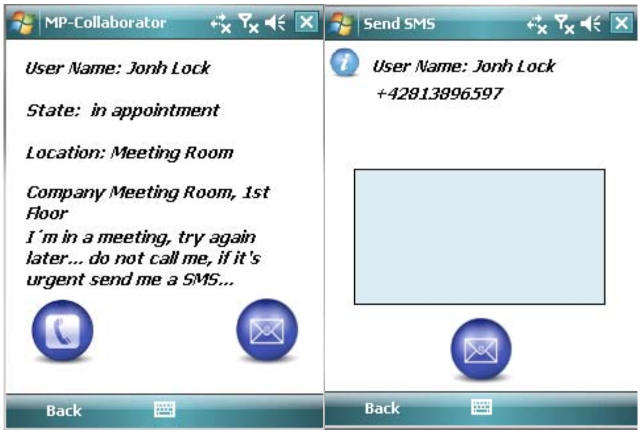
Fig. 4. MP-Collaborator primary window.

The Menu *softkey* offers extended functionality such as application termination, user search, application help and about information, which beside the closing application button will offer a search function, an about and a help file. Every time the system finds another connected application in the network; this one identifies the user, separating users by groups. As a result, it is easier to look up in the application interface for an already connected user. A group may be a department or a team inside the organization, for example. Users that are not connected, stay in the 'Offline' group (corresponding to the offline users); while the other connected users are separated by groups, and also placed in an 'Online' group (or state).

Users can also be found through the "search collaborator" function in the primary window main "Menu". This is useful when the list is too large. The main Menu also provides the application general information in the about, and a help file that essentially presents the application user manual.

The application will use the *Device ID* number to identify a PDA and associates it to its user, together with the interface IP. The application possesses an automatic and manual refresh system that is used to refresh and update all the transmitted data between users and verify which users are online and offline. By selecting a remote user, a new window will appear (part (a) of figure 5). This window displays, from top to bottom, user name, user profile (state, location, and if possible Pocket Outlook Calendar data), and finally the call and SMS buttons.

A dedicated window is used to send an SMS, as part (b) of figure 5 depicts. If the remote user is busy in a meeting a call cannot be made, so MP-Collaborator reacts accordingly. Also, if the remote user does not discloses his mobile contact number, call or SMS is not available.



User Name: Jonh Lock

State: in appointment

Location: Meeting Room

Company Meeting Room, 1st Floor

I´m in a meeting, try again later... do not call me, if it's urgent send me a SMS...

User Name: Jonh Lock

+42813896597

Fig. 5. MP-Collaborator (a) profile window and (b) SMS window.

All user information about location and profile will be stored in an internal database on the mobile device, eventually been replicated across devices. Finally, the application will be able to access the Windows Mobile Pocket Outlook, and verify if a user is mentioned in an appointment. This connection allows access to contextual information, such as, the appointment location, the time that it stars and ends, and the subject of the appointment.

With this network architecture no dedicated central server is needed to store any information, since information is distributed among the several connected devices. This architecture is indeed a peer-to-peer architecture with server-less operation, where in the same application the "server" and "client" roles are played in "parallel".

B. Application validation

This section presents some tests issued with MP-Collaborator, validating our approach. The tests were also performed in design time, to collect debug information and change as necessary.

MP-Collaborator validation was performed using exhaustive running tests. We divided testing into two fundamental areas: general application functionality and networking tests. For all tests we used real devices, instead of the bundled windows mobile 6.0 emulator, as shown in Figures 6, 7 and 8.

General application functionality tests focused local application functionality, like database testing, search engine testing, and stability issues, among others. This kind of testing enabled several debugging operations. For this testing we also thoroughly tested GPS functionality and Pocket Outlook Calendar connection. Although difficult to attest if MP-Collaborator is bug-free, the number of days the application was put through its paces allows a strong degree of confidence.

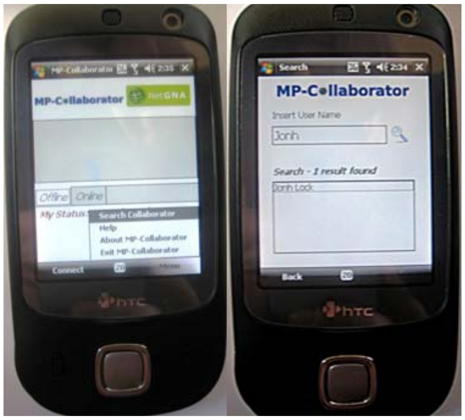


Fig. 6. Two screens of MP-Collaborator in a real device: main menu and search engine.

The second area of testing involved the validation of network-related features. Several devices were used, namely from HTC and Asus. We tested several scenarios, from communication establishment, communication disruption, and profile change. We used Wi-Fi with fixed IP addresses, using TCP/IP sockets.

Communication dynamics is imposed by refresh time, since each refresh triggers network access. In the current MP-Collaborator implementation, refresh time is 18 seconds. As a result, user profile change takes at most the refresh time to be updated on the clients. We verified this assumption by disconnecting users and changing user's profile.

We also tested communication establishment under worst-case scenarios, like the second user connecting one second after the first one. As expected in this worst-case scenario, the first user only "sees" the second user on the first refresh and not on application start.

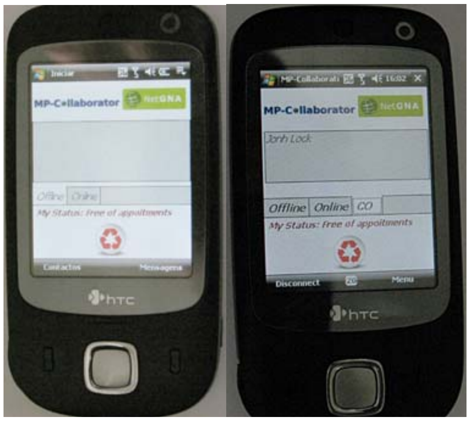


Fig. 7. Michael Long PDA – (a) MP-Collaborator starts and (b) with John Lock remote user.

We now explain a real scenario with three users – Michael Long, John Lock and Ben Linos. Initially they are all disconnected. Michael starts connection and does not see the other two users, as Figure 7 (a) depicts. Then John Lock starts MP-Collaborator and after a while Michael sees him (figure 7 (b)). When Ben Linos jumps in, Michael sees both users, as presented in Figure 8.

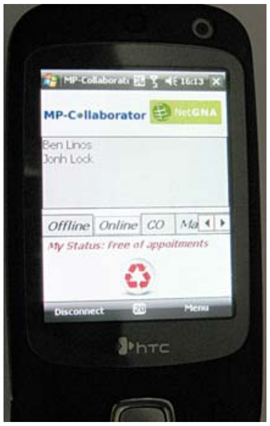


Fig. 8. Michael Long PDA after Ben Linos connection.

V. Conclusions

This paper presents a mobile solution that optimizes the communication between collaborators in an organization. We developed a ubiquitous location and context-aware mobile application that helps the user in making communication decisions. Any user knows the location and context of all other connected users, the source of the location and context-aware information is the Location Based Service installed in the device that crosses the position with the application database, or the data that the application returns from de Windows Mobile Pocket Outlook Calendar. Any user without knowing any configuration or knowledge of the application technology can perfectly use the MP-Collaborator, almost without reading the user's manual.

This mobile application may benefit from some improvements including the introduction of remote automatic appointment requests, client discovery through mobile-IP by enabling auto-configuration. The user interface can be further improved by including the feature to search a collaborator inside the main form.

Ackowledgements

Part of this work has been supported by the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, the Euro-NF Network of Excellence of Seven Framework Program of EU and the Natural Science and Engineering Research Council of Canada (NSERC).

References

[1] J. E. Bardram and C. Bossen, "Mobility Work: The Spatial Dimension of Collaboration at a Hospital", *Comput. Supported Coop. Work*, ISSN: 0925-9724, vol. 14, No. 2,, pp. 131-160, 2005.

[2] L. Borrmann and F. N. Paulish, "Software Architecture at Siemens: The Challenges, our Approaches, and Some Open Issues", in *First Working IFIO Conference on Software Architecture*, San Antonio, Texas, USA, 1999.

[3] T. Berners-Lee and R. Cailliau, "The World Wide Web", *Communications of the ACM*, vol. 37, No. 8, pp. 76-82, 1994.

[4] B. Beurer-Zuellig and M. Meckel, "Smartphones Enabling Mobile Collaboration", in *41st Annual Hawaii International Conference on System Sciences*, ISBN: 978-0-7695-3075-8 Waikoloa, Hl, 2008, pp. 49-58.

[5] S. Berger, R. Mohr, H. N osekabel, and K. J. Schafer, "Mobile Collaboration Tool for University Education", in *Proceedings of the Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE '03)* Linz, Austria: IEEE Computer Society, pp. 77-78, 2003.

[6] P. Neves, D. Ferreira, D. Esteves, D. Felix, and J. Rodrigues, "InHand - Mobile Professional Context and Location Aware Tool", in *16th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2008)*, , pp. 80-84, September 2003

[7] V. Roussev, G. P. Priego, and G. G. Richard, "TouchSync: Lightweight Synchronization for Ad-Hoc Mobile Collaboration", in *International Symposium on Collaborative Technologies and Systems (CTS 2006)*, Las Vegas, NV, USA, pp. 181-188, May 2006.

[8] M. Weiser, "The World is not a Desktop", *interactions*, ISSN: 1072-5520, vol. 1, pp. 7-8, No. 1, 1994.

[9] M. Weiser, "The Computer for the 21st Century", *Scientific American*, vol. 265, No. 3, pp. 94-104, 1991.

[10] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges", *IEEE Personal Communications*, ISSN: 1070-9916, vol. 8, No. 4, pp. 10-17, August 2001.

[11] L. Jessup and D. Robey, "The Relevance of Social Issues in Ubiquitous Computing Environments", *Communications of the ACM*, ISSN: 0001-0782, pp. 88-91, 2002.

[12] P. Bellavista, A. Kupper, and S. Helal, "Location-Based Services: Back to the Future", *IEEE Pervasive Computing*, vol. 7, No. 2, pp. 85-89, 2008.

[13] P. Rotter, "A Framework for Assessing RFID System Security and Privacy Risks", *IEEE Pervasive Computing*, ISSN: 1536-1268, vol. 7, No. 2, pp. 70-77, 2008.

[14] W. Ho, A. Smailagic, D. P. Siewiorek, and C. Faloutsos, "An Adaptative Two-Phase Approach to WiFi Location Sensing", in *4th annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW 2006)*, Pisa, Italy, pp. 452-456, March 2006,

[15] J. Sharp, *Microsoft Visual C# 2005 Step by Step*: Microsoft Press, 2005.