

Developing Collaborative Peer-to-Peer Applications on Mobile Devices

Cláudio Sapateiro¹, Nelson Baloian², Pedro Antunes³, Gustavo Zurita⁴

¹ *Systems and Informatics Department, Polytechnic Institute of Setúbal, Portugal*
csapateiro@est.ips.pt

² *Department of Computer Science, University of Chile, Santiago de Chile, Chile*
nbaloian@gmail.com

³ *Informatics Department, Faculty of Sciences, University of Lisbon, Portugal*
paa@di.fc.ul.pt

⁴ *Management Control and Information Systems Department, Business School, Universidad de Chile, Chile*
gnezurita@fen.uchile.cl

Abstract

Mobile computing has experienced a rapid growth and there are nowadays many applications being developed. According to the literature, many of these applications are designed for collaborative work or learning and share the same requirements: they implement a peer-to-peer communication architecture, there are subgroups inside the group of participants which have to work together and coordinate with the rest of the groups and they need to implement a proper human-computer interaction paradigm since they have a reduced area to interact with the application's interface. It is then quite natural to think about the development of a framework supporting this kind of applications. In this paper we present such a framework and an example of its application.

Keywords: mobile computing, collaborative computing, middleware, peer-to-peer architectures.

1. Introduction

Mobile and pervasive computing has seen a rapid development in the last years. Portable, handheld computing devices are getting more popular as their capabilities increase. Therefore, people having the need to work on-the-field have now the possibility to be supported by computer technology. The ad-hoc networking capability of handheld devices enables the development of supporting tools for collaborative work anywhere, anytime.

In the past years, we have seen an explosion of new collaborative work supporting systems for PDAs, smartphones and other mobile devices. Applications allowing users to collaborate in real time using wireless connected mobile devices building ad-hoc networks have attracted the attention of many authors. Some of the scenarios for which these applications have been developed are the following [2]:

- Educational activities involving group of students and teachers in collaborative room environments
- Mobile devices have been used to mediate between healthcare personal or to support the relationship between therapists and patients in a hospital
- Group of people attending a meeting can share ideas and data by means of their mobile devices
- Field survey operations in remote areas with no fixed infrastructure can be easily facilitated.
- Members of an organization can register and share their knowledge about important processes in a more flexible and informal way with a mobile knowledge management system.

Mobile computing can offer communication and data processing in scenarios requiring high mobility, users may come and go in the middle of a working session, and working sessions may start and end without preparation. Also a common characteristic of these scenarios is that they require an easy and swift way to interact with the computing device.

Because of their physical characteristics, mobile devices impose some restrictions which have to be taken in account when designing the application [4]. Some of them are low-bandwidth and high latency, low processor power, small display size, short battery life and limited input methods.

Since the early days of distributed, collaborative applications development many authors recognized the need for middleware in order to ease the programming of this kind of software, each one fulfilling different requirements [22], [17], [15]. They differ in distribution schemes of the shared data, communication mechanisms, and application architecture they support. In most cases, middleware for developing collaborative applications addresses the problem of communication between the applications separately from the human-computer interaction and the problem of the formation and management of user groups. However, in the mobile scenario, these three issues are tightly coupled. In fact,

the type of data the application's interface allows the user to input will determine the data structure that will be necessary to transmit between the applications. Also, the dynamic formation of groups and sub-groups of users will determine which are the applications having to share their data that at a certain moment. In this paper we present a framework called MCDeveloper that supports the development of peer-to-peer applications for scenarios where there is a fast change of users and the input information consists of mainly free hand sketching.

The next section introduces the developed framework. Section 3 describes one of the considered application domains for the framework instantiation: a mobile application to assist IT service desk teams when facing infrastructure failures. The conceptual model of the considered application domain and its implementation are described in section 4. The experiments conducted with domain experts, to evaluate both the proposed conceptual approach and its implementation, are described in section 5.

2. The Framework

As we said, the framework will support three aspects of the application since they are highly related in scenarios where mobile computing is required. These are communication, group management and data input. The middleware we developed consists of a set of classes implementing an API the programmer can use. These classes are available in Java and C# and implement the necessary mechanisms for converting data objects from their internal representations into an XML representation, transmit them across platforms and convert the XML representation into the corresponding internal one. They also provide classes for implementing free hand sketching and recognizing gestures.

2.1 Communications

In many mobile scenarios, the only available network will be the mobile ad-hoc network (MANET) provided by the networking capabilities of the mobile devices. This means that the communication and the data architecture must follow a peer-to-peer schema avoiding a central server keeping the "master" copy of the data and/or the list of active users.

In order to have an application join the group of active partners in the ad-hoc network it has to instantiate an object of the Node class. This will start a service which will send multicast messages at regular intervals to the group in order to inform other participants of the presence of a new participant. It will also start consuming multicast messages from other partners present in the ad-hoc network. This allows the application to maintain a list of active participants

updated. Every time a multicast message of a new participant is received, its ID and IP number are stored in the list and a TCP/IP connection is established with that application through which data will be shared.

The data sharing mechanism is based on a "shared objects" principle. A shared object is an abstract class which should be extended in order to create an object class whose state will be transmitted to all active participants when the object changes its state, this is when one or more variables change their value. The programmer implements a shared object by extending the SharedObject abstract class provided by the framework. Apart from declaring the field variables and methods for this object, it is often necessary to implement a method called postProcess which will be called every time the object state is updated by a remote application.

2.2 Group Management

The mobile collaborative working scenario supported by ad-hoc networks motivated us to for developing this middleware because of the need to have applications implemented and running in different platforms to share data. In this scenario, we also recognized the need to have the possibility of defining subgroups of partners inside the whole group of active participants. In order to support this issue, the middleware implements mechanisms that define user groups of applications that he/she can join and leave. This information is stored in the communication node and is used when the copy of an updated object has to be distributed among participants. Accordingly, the communication mechanism will send updated shared objects only to applications belonging to the group. An application can join more than one group.

2.3 Human-Computer Interaction

According to [7] a handheld application interface must imitate the pen-and-paper metaphor so users can interact naturally with the computer in varied situations. A pen-based system offers natural and intuitive interface less disruptive of creative processes involving various designers, helping users capture ideas and engage in collaborative modeling.

A good pen-based interface should make extensive use of gestures. In a survey [16], it was found that the most frequent actions were deleting, selecting and moving, and that users consider these actions to be efficient as a form of interaction, as well as convenient, easy to learn, utilize and remember, and potentially an added advantage for the interface.

A powerful means of supporting interpersonal communication is sketching [13]. Face-to-face communication involves the use of diagrams and drawings in a medium that permits users to share views

while they converse, a process that helps eliminate ambiguities and rapidly communicate new or complex ideas. The framework supports the development of interfaces that imitate the pen-and-paper metaphor based on gesture recognition and sketching by offering a library of classes which already implement these functionalities. These can be easily modified by extending these classes, in order to adapt them to the application being developed.

3. Application Example

The work reported in this section describes the development of an application to assist IT Service desk teams in addressing critical non routine situations. A number of situations (e.g. servers, services, and network links failures) are perceived as critical to organizations in which business processes heavily rely on IT infrastructure. Despite the existence of procedures to address some recurrent problems, two main issues may strongly condition the effectiveness of the problems handling strategy. First, standard procedures may not accommodate the specific contextual/contingent factors of the situation; second, many of the misses in procedures can be solved by using tacit knowledge and experience of the involved participants, to develop solutions or temporary workarounds.

The application's purpose is to assist IT service desk teams in such non routine situations under time pressure and work activities are in unplanned mode and may lack overall consistency and coordination efficiency.

To elicit current critical situations and existing practices, we conducted a set of interviews to two IT service desk teams from two organizations. The first team was constituted by one team coordinator, two senior technicians, and two junior technicians; the second team consisted in one team coordinator, one senior technician and one junior technician. Regarding the critical incidents, most serious cases were related with server failures, mostly due to disk failure, and connectivity losses in some network segments compromising a wide variety of services. The existing preventive practices rely heavily in monitoring the active network elements through a control panel fed by SNMP messages, where alerts are displayed and emailed to the technicians. Several equipments are under SLA agreements with suppliers and a spare stock exists. Actual diagnosis and recovery practices rely heavily in the field experience of each team member. The collaboration process addressing critical situations is essentially supported by quick informal meetings, phone calls and chat tools. A key expressed concern was the need to document this kind of interventions for future reference although trouble ticket software is available, it is only used (sometimes) for incident opening and some (few) occasional post mortem annotations to close it, due that no mobile module is

available and technicians have to be in the desktop to interface with it.

From the compiled interviews' results we established a set of system requirements grounded on several groupware research proposals such as: heuristic evaluation [20], knowledge-based evaluation frameworks [22], teams' performance [1] and situation awareness [19]. Table 2 presents the list of requirement which was presented to IT service desk teams to rate their relevance regarding the specific scope we were addressing. The ratings scale was from 1 to 4 in which: 1 - Not perceived as important, 2 - Less important, 3 - Important and 4 - Very important.

The ratings emphasize the requirements 1, 3, 4, 5, 6, 7, 8, 13, 14 as Important or Very Important. Requirement 2 was considered Not Important. And requirements 9, 10, and 1 were not consensual. A more detailed analysis of no consensual results in conjunction with the recorded interviews yield the following considerations: Knowledge transfer and incident documentation revealed Very Important to teams' leaders; situation representation and knowledge externalization support revealed important to more junior technicians.

Table 1. Requirements

| N° | Requirement | Influence Area |
|----|---|----------------------|
| 1 | Communication support through shared artifacts | Groupware |
| 2 | Transitions between individual and team work | |
| 3 | Facilitate situation (specific issues) monitoring | |
| 4 | Minimal overhead work demand | |
| 5 | Mobile end device availability | |
| 6 | Assist situation context understanding | Situation Awareness |
| 7 | Perceive who is involved | |
| 8 | Assist situation size up | |
| 9 | Assist (overall) situation representation | Knowledge Management |
| 10 | Knowledge externalization support | |
| 11 | Knowledge transfer support | |
| 12 | Incident handling documentation | Performance |
| 13 | Improvement in diagnosis time | |
| 14 | Improvement in recovery time | |
| 15 | Number of coupled incidents simultaneously attended | |

4. Developed Prototype

4.1 Conceptual Model

Considering the Swiss-cheese model for organizational accidents proposed by Reason [18] we propose that different organizational dimensions should be easily aligned to establish barriers for accidents progressions. In extending such model to real time operations we've adopted a phenomenological perspective on context of action characterization, traditionally used in social sciences, which regards contexts as relational entities relating all involved actions and objects, and evolving dynamically as actions unfold [3]. Such approach is consistent with the studies of naturalistic decision making which reveals that under extreme conditions, the decision process is deeply influenced by situation context and experience of involved actors [13].

The proposed approach characterizes the incident situation as a set of Situation Dimensions (SD) that should be easily correlated through a set of shared artifacts: Situation Matrixes (SM), which will help to establish situation context and coordinate actions.

Samples of SD can be: involved actors, goals, proposed actions, resources, etc. These dimensions will be correlated through SM like actor vs. actions, goals vs. actions, resources vs. actors, etc. The initial set of both SD and corresponding SM should be defined by domain experts. Nevertheless, as situation evolve additional SD can be identified and correlated with existing SD through new SM.

Editing both SD and SM is done collaboratively on a real time basis. The proposed collaboration model is grounded in resilience engineering principles [12], emphasizing redundancy regarding existing IS, power deference to expertise among involved actors, situated action and shared situation awareness establishing. With the proposed approach we aim to capture individual contributions to the overall situation understanding and handling promoting the externalization of actors' tacit knowledge and experience to enhance collective sensemaking [27] and situated framing [8].

Figure 1 depicts the proposed model. The correlation of different SD values may use different signs, with different sizes and/or colors to establish the semantic meaning of the relation. This semantic is defined by the application domain experts.

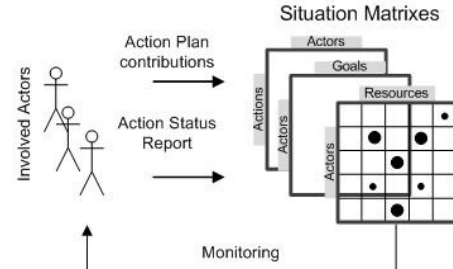


Figure 1: Proposed model

4.2 Application

In this section we present the developed application to instantiate the proposed model, over the previously introduced platform.

Attending constraints of the considered application domain, the work overhead imposed by the use of the proposed application should be minimal and its user interface very simple.

By drawing a “half rectangle” on the screen the gesture will be recognized by the system as the creation of a new SM (Figure 2a). A list of available SD will be displayed and the SM may be populated by dragging them to the rectangle representing the SM, as shown in figure 2c).

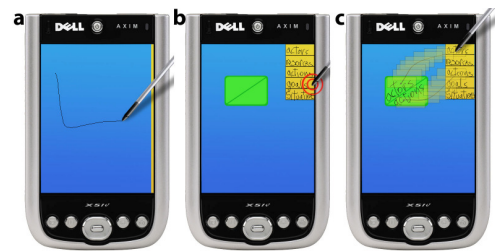
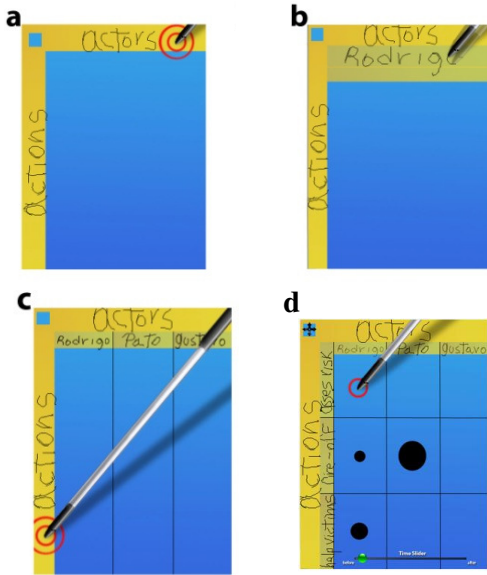


Figure 2: PDA Prototype: a.SM Creation b.SD c.Assigning SD to SM

In order to create a new column the user has to double click on the column label (Figure 3a). After this, the user has to enter the header text for the column as shown in the Figure 3b. Figure 3c shows that the user has already created 3 columns and is starting the creation of a new row using a similar procedure.

Since different users may be interested in viewing different parts of the matrix according to their context of action. Therefore, the system allows hiding rows or columns. The hidden row or column will be represented by a thicker line. In order to show again a hidden column or row the user has to double click on the thick line. Regarding the SM navigation capabilities, the left-right and up-down scrolling functionality, combined with zoom-in and zoom-out functionality, enables an easy and swift traversing of the whole matrix.



**Figure 3: PDA Prototype: a.SD definition
b-c.insert item in SD d.relate SD items**

5. Evaluation

Collaborative systems evaluation raises many methodological concerns that have received attention from researchers in CSCW community. Evaluation strategies may differ in: moment (design, prototype, finished product), time span (hours, weeks, months, years), local (laboratory, work context), people involved (domain experts, final users, developers), type (quantitative, qualitative) [11]. Also the scope of the evaluation process may target different dimensions of the proposed system, from technical dimension (e.g. interoperability) to organizational impact dimension (e.g. effects on tasks performance) [9] [23]

Typical evaluation strategies consist in field methods in actual context with real users and inspection techniques. Although field methods allow capturing more realistic problems and requirements, they could be difficult to settle due several reasons: time investment, scenario setting, associated costs, prototype maturity, etc. Inspection techniques allow, to some extent, to overcome some of these shortcomings since they are much less costly than field methods, and they can often be used earlier and more frequently in the development cycle. In [20] the authors indicate that it is possible to contextualize inspection techniques through the use of work scenarios jointly constructed with domain experts and those techniques may lead to many of the same problems that were found in field studies. Nevertheless, the authors defend that after a number of low cost evaluations, when possible and/or in later maturity states of the prototype, field approaches should be complementarily used, which may help to prioritize the issues identified through the inspection techniques.

We based our evaluation methodology in discount inspection techniques recurring to a scenario based approach collaboratively constructed with domain experts [5] [10]. We conducted a workshop in which the proposed approach was discussed under the context of scenario, previously elaborated with team's senior members.

The discussion of the current and desirable strategies to address the proposed situation, reveal that when unplanned situations emerge team members experience strongly constrains the situational shared awareness. The proposed collaboration and situation representation model were perceived as relevant to externalize existing tacit knowledge structures, establish shared situation awareness and document the situation. The set of SD proposed by the teams were: Equipments, Actors, Locations, Actions and Activities, which should be correlated in the following suggested as relevant SM:

- Actions-Steps, detailing operational activities (e.g., check router X, reboot switch Y).
- Actors-Steps, defining responsibilities.
- Equipment-Actors, expressing the persons responsible for the equipment (e.g., who is empowered to activate a supplier warranty, who is habilitated to inspect a Linux server or a specific service).
- Equipments-Locations, allowing team members (mostly junior) to know the equipment locations (e.g., main gateway of building C6 is located in room 6.3.0.1).

Regarding application usability, some additional improvements were suggested. A better support to navigation between existing SM it will be much appreciated. Also to keep the situation information up to date without adding unacceptable overhead, presents major challenges. For instance, status reports and situation assessments are hard to track due to their dependencies on the explicit user interactions. We pretend to improve the developed prototype by minimizing interactions and explicit user's information declaration by for instance, using a pulling strategy: as information becomes old, respective users may be prompted to report their validity, in combination with a visualization schema to express the degradation of the quality of the available information.

6. Conclusion

In this paper, we presented a framework which provides a basis to ease the development of mobile applications supporting collaborative work in peer-to-peer environments and an example of an application developed over it. We observed that the programmers invested little time in learning the usage of the framework and the development took much less time compared with the time necessary to build a similar application from the scratches. For this application, all

the components of the framework were used, showing that the features it provides are highly necessary in such a scenario. Currently, new applications are being developed and tested using the framework. First informal tests show the framework is highly appreciated by programmers and the resulting applications are robust and effective.

We are also further studying alternative user interaction modes trying to minimize the dependence on explicit user reporting which constitutes an overhead work that should be, whenever possible, avoided. We are studying a pulling strategy to handle timeliness: 1) when users input information, a deadline is also introduced and when this expires users are prompted to report information validity; 2) if no deadline is introduced, then the specified correlation will incrementally become more visually transparent.

Two other issues are also being considering in the future releases of the prototype: the first concerns the possibility to have icons relating the SD to express possible different semantics of that relation; the second regards with a notification mechanism that will allow alerting to specific users to some specific relation of the situation. This notification mechanism can be extended to mail notification in order to involve other organizational levels if the situation escalation imposes.

Acknowledgement

This paper was partially supported by: Fondecyt 1085010 and the Portuguese Foundation for Science and Technology, Project FCT (PTDC/EIA/67589/2006).

References

- [1] Baeza-Yates, R. and J. Pino (2006). "Towards formal evaluation of collaborative work and its application to information retrieval." *Information Research* 11(4): 271.
- [2] Baloian, N., Zurita, G., Antunes, P., Baytelman, F. (2007): "A Flexible, Lightweight Middleware Supporting the Development of Distributed Applications across Platforms". *Procs. Of CSCWD-07*, Melbourne, Australia, April, 2007, IEEE press, pp. 92-97.
- [3] Borges, M. R. S., P. Brézillon, et al. (2004). "Groupware system design the context concept" *Proceedings of the CSCWD*.
- [4] Buzko, D., Lee, W., and Helal, A. "Decentralized ad-hoc Groupware API and framework for mobile Collaboration",
- [5] Carroll, J. M. (2000). *Making Use: scenario-based design of human-computer interactions*, MIT Press.
- [6] Cox, T., Perdomo, J. and Thabet, W. "Construction field Data Inspection Using Pocket PC Technology", *International Council for Research and Innovation in Building and Construction*, CIB w78 conference, 2002.
- [7] Dai, G. and Wang, H. "Physical Object Icons Buttons Gesture (PIBG): A new Interaction Paradigm with Pen", *LNC3 3168*, 2005, pp. 11-20
- [8] Gasson, S. (2004). *A Framework For Behavioral Studies of Social Cognition In Information Systems*. ISONeWorld.
- [9] Gauducheau, N., E. Soulier, et al. (2005). *Design and evaluation of activity model-based groupware: methodological issues*. 14th International Workshops on Enabling TEchnologies: Infrastructure for collaborative enterprise (WETICE), IEEE Computer Society.
- [10] Haynes, S., S. Purao, et al. (2004). *Situating evaluation in scenarios of use*. CSCW, ACM.
- [11] Herskovic, V., J. A. Pino, et al. (2007). *Evaluation Methods for Groupware Systems*. CRIWG.
- [12] Hollnagel, E. and D. D. Woods (2006). *Resilience Engineering Precepts*.
- [13] Klein, G. (1998). *How people make decisions*. Cambridge, Massachusetts, MIT Press.
- [14] Kenneth, D., Forbus and Usher, J. "Towards a computational model of sketching", *Proceedings of Intelligent User Interfaces*, 2001, pp. 77-83.
- [15] Litu, R., and Zeitoun, A.: "Infrastructure support for mobile collaboration", *Proceedings of the HICSS*, 2004, Hawaii, USA.
- [16] Long, A., Landay, J. and Rowe, L. "PDA and gesture Use in Practice: Insights for Designers of Pen-based User Interfaces", <http://bmcrc.berkeley.edu/research/publications/1997/142/clong.html>
- [17] Lukosch, S: "Adaptive and Transparent data Distribution Support", *Proceedings of the CRIWG'02 conference*, September, La Serena, Chile, 1002, pp 255-274.
- [18] Reason, J. T. (1997). *Managing the risks of organizational accidents*, Aldershot: Ashgate.
- [19] Steves, M. P., E. Morse, et al. (2001). *A comparison of usage evaluation and inspection methods for assessing groupware usability*.
- [20] Urnes T. and Nejabi, R.(2004) "Tools for implementing groupware: Survey and evaluation", *Technical Report No. CS-94-03*, York University, 1994.
- [21] Vizcaino, A., M. Piattini, et al. (2005). *Evaluating collaborative applications from a knowledge management approach*. 14th IEEE International Workshops on Enabling Technologies: Infrastructure for collaborative enterprise WETIC'05.
- [22] Vyhmeister, R., P. R. Mondelo, et al. (2006). "Towards a Model for Assessing Workers' Risks Resulting from the Implementation of Information and Communication Systems and Technologies." *Wiley InterScience* (www.interscience.wiley.com).
- [23] Weick, K. E. (1996). *Sense Making in Organizations*, Sage Publications.
- [24] Yang, L., Landay, J.A., Guan, Z., Ren, X. and Dai, G. "Sketching Informal Presentations", *Fifth ACM International Conference*