*Journal of Universal Computer Science, vol. 17, no. 2 (2011), 183-202*
*submitted: 17/9/09, accepted: 7/12/10, appeared: 28/2/11 © J.UCS*

# The Iceberg Effect: Behind the User Interface of Mobile Collaborative Systems

**Valeria Herskovic**
(Department of Computer Science, Pontificia Universidad Católica de Chile, Santiago de Chile Chile
vherskov@ing.puc.cl)

**Sergio F. Ochoa**
(Department of Computer Science, Universidad de Chile, Santiago de Chile, Chile
sochoa@dcc.uchile.cl)

**José A. Pino**
(Department of Computer Science, Universidad de Chile, Santiago de Chile, Chile
jpino@dcc.uchile.cl)

**Andrés Neyem**
(Department of Computer Science, Pontificia Universidad Católica de Chile, Santiago de Chile Chile
aneyem@ing.puc.cl)

**Abstract:** Advances in mobile technologies are opening new possibilities to support collaborative activities through mobile devices. Unfortunately, mobile collaborative systems have been difficult to conceive, design and implement. These difficulties are caused in part by their unclear requirements and developers' lack of experience with this type of systems. However, several requirements involved in the collaborative back-end of these products are recurrent and should be considered in every development. This paper introduces a characterization of mobile collaboration and a framework that specifies a list of general requirements to be considered during the conception and design of a system in order to increase its probability of success. This framework was used in the development of two mobile collaborative systems, providing developers with a base of back-end requirements to aid system design and implementation. The systems were positively evaluated by their users.

**Keywords:** Mobile shared workspaces, hidden collaboration requirements, requirements for communication and coordination.
**Categories:** D.2.1., D.2.2

## 1 Introduction

Advances in mobile computing and wireless technologies, coupled with decreasing mobile device prices, are opening new possibilities to use these technologies to support various collaborative activities. Some scenarios that may benefit from these technologies are the manufacturing industry, construction, agriculture, education and generally, any activity involving people working collaboratively while on the move.

Mobile collaborative systems allow users to work in a loosely coupled way in order to accomplish their goals [Pinelle and Gutwin 2006]. The development of these systems is a complex and not well-understood activity. However, there are several requirements that are present in most mobile collaborative systems, which reduce the complexity of the development process. We refer to these recurrent needs as "*general requirements*". They are usually related to background processes in charge of providing the supporting mechanisms to enable collaboration in a mobile work scenario. Unfortunately, these general requirements are not visible for most users and developers. Therefore, developers could fail to consider these requirements in the development process or they could be included late, jeopardizing the project success.

In order to explain this situation in depth, let us consider the basic architecture of a collaborative system. It is clear that collaboration requires communication and coordination [Ellis et al. 1991]. Requirements related to these concerns should be layered (Figure 1). Usually, requirements involved in the upper layer are highly visible to users and developers, because these needs are mainly related to the application front-end. They describe the functionalities that a specific application has to expose to end-users. For example, in a workspace supporting students taking notes during lectures on Tablet PCs, these requirements will be related to how annotations are shared and whether other users can edit them or not.

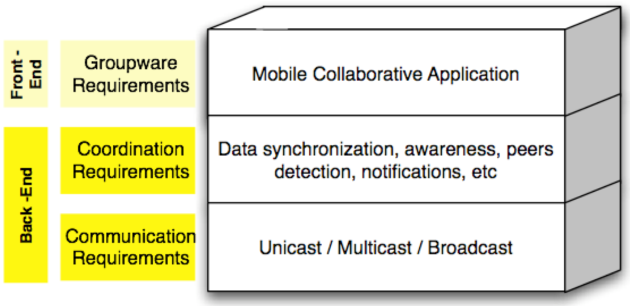


*Figure 1: Mobile collaborative system architecture*

Requirements related to the coordination and communication layers correspond to functionality that is part of the application back-end. Examples of these requirements are user autonomy, peer dynamic discovery, service and information interoperability, data synchronization, and broadcast messaging. Clients and developers are typically unaware of the issues, and thus these requirements become invisible for them. These needs are recurrent in mobile collaborative systems and they mainly involve the communication and coordination layers that are the basis for mobile collaboration. Since these general requirements are difficult to identify, no traditional elicitation processes can be used to record them.

These recurrent requirements produce a gap between the software required by the client and the final product. We call this gap the *Iceberg Effect*: only a few requirements are visible to stakeholders, while most requirements, including the most complex ones, are hidden (Fig. 2). This makes development a challenging process, in which the final product may not be successful in its intended use scenario.

This paper presents a framework that specifies a list of general requirements to be considered during the conception and design of a mobile collaborative system in order to increase its probability of success. The framework was obtained based on a survey of relevant research as well as from the authors' experiences developing mobile collaborative systems.

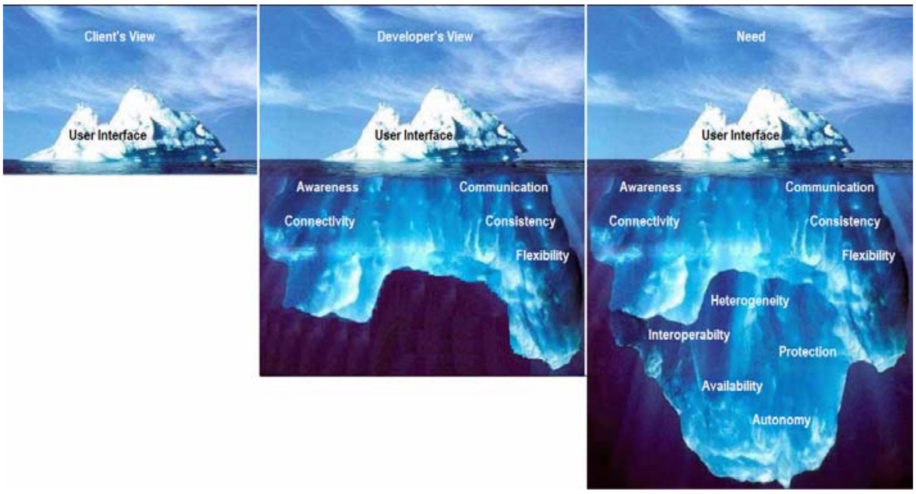


*Figure 2: Iceberg Effect in mobile collaborative software requirements*

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 introduces a characterization of how mobile collaborative work takes place. Section 4 details the list of general requirements. Section 5 illustrates how these requirements were considered in the development of two mobile collaborative applications. Finally, section 6 presents the conclusions and future work.

## 2 Related Work

Computer-supported mobile collaboration is a branch of Computer-Supported Cooperative Work (CSCW) emerging from the new opportunities brought by wireless communication technologies and mobile computing devices. The result is a new collaboration scenario where most of the traditional solutions to typical design issues are not applicable [Neyem et al. 2007].

Several authors have studied mobile collaboration, analyzing the characteristics that distinguish it from traditional collaborative systems, and extracting requirements commonly involved in this type of interaction. Requirements are usually obtained from particular experiences developing mobile collaborative applications or may be extracted from theoretical models that explain how users collaborate. Some authors focus on just one or two requirements, such as awareness or communication [Carter et al. 2004, Menchaca-Mendez et al. 2004]. The most complete studies were recently presented by Pinelle et al. [Pinelle and Gutwin 2006] and Neyem et al. [Neyem et al. 2008].

Roth encountered seven challenges when developing QuickStep, a platform to support developers of mobile collaborative applications and Pocket DreamTeam, the mobile version of the DreamTeam groupware platform [Roth 2002]. These challenges are the following: communication, coordination, architecture, data distribution and consistency, user interfaces, security and privacy, and realization issues. Kortuem et al. identified challenges such as privacy, security and resource discovery, motivated by a scenario of impromptu MP3 file sharing [Kortuem et al. 2001]. From these challenges we may extract requirements that pertain to mobile collaborative systems.

Other authors who have developed mobile applications have listed several requirements that all mobile collaborative systems should consider. Tietze developed components as a basis for building systems, listing both end-user and developer requirements for them [Tietze 2001]. Some end-user requirements are the access to shared artifacts, computer guidance in selecting appropriate tools, provision of group awareness, and support for mobile work. Divitini et al. outline some functional requirements for UbiCollab, a ubiquitous shared workspace to support mobile users. Some of the most important requirements are the following ones: awareness, several communication channels, and transitions between collaboration modes [Divitini et al. 2004]. Parsons et al. identified generic mobile environment issues such as mobility, mobile interface design and communication support as key qualities of collaborative mobile learning [Parsons et al. 2007]. Cheverst lists a set of requirements based on the power distribution industry's requirements [Cheverst 1999], such as the ability to operate in a heterogeneous networking environment and the ability to support field engineer mobility. Zurita et al. describe requirements for a mobile application involving PDAs to support meetings [Zurita et al. 2008] and learning activities [Zurita et al. 2007].

Another perspective is provided by authors who propose models and architectures for mobile collaboration, which we may study to find the common back-end requirements. Essman and Hampel use the CCC (Context, Connectivity, and Consistency) model to characterize mobile collaboration and propose a four-tier architecture. From this model we may extract several requirements, such as the need to identify the work context and define a replication and caching strategy [Essman and Hampel 2005].

The reviewed literature provides many perspectives on the important aspects to consider when developing mobile collaborative systems. However, most of these studies are focused on a specific area of application and do not take into account the impact of general requirements on the development project success, nor how to relate the requirements to how mobile work takes place. This paper aims to provide a more comprehensive view of typical mobile collaboration requirements and the relationships among them. The next section characterizes mobile collaborative work and it presents a classification of the interaction scenarios between two users.

## 3 Mobile Collaboration Characterization

Participants in a mobile collaboration process must be able to work autonomously through the system [Essman and Hampel 2005, Munson and Dewan 1997, Neyem et al. 2007, Monares et al, 2011, Pinelle 2004, Pinelle and Gutwin 2005], storing their work in an individual workspace. In mobile collaboration, the work is usually loosely coupled [Churchill and Wakeford 2001], so mobile users will work autonomously most of the time and they will carry out sporadic on-demand collaboration processes (Fig. 3). After engaging in collaboration, users will return to autonomous work.

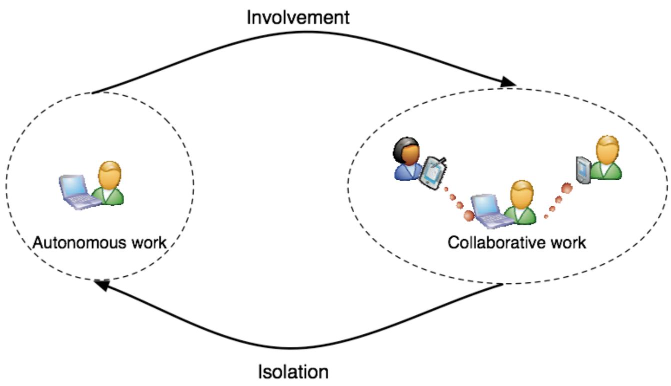


*Figure 3: Work in mobile collaboration*

We will characterize the periods of collaboration between actors as a way to clarify the required support. Collaborative systems are usually organized in a space/time matrix [Dix et al. 1998], which would be difficult to apply to mobile collaborative systems, since in this scenario physical location is constantly changing, so the line between remoteness and co-location becomes blurred. Therefore, we prefer to model the collaboration scenarios in terms of simultaneity and reachability.

We define two actors are *reachable* if they are *able to exchange information and communicate in a highly predictable way*, i.e., there is an available communication channel that allows them to communicate and expect a response in a certain period of time [Grudin 1994]. We define two actors are *simultaneously present* if they are *available to work synchronously.*

Considering the dimensions of simultaneity (simultaneous and non-simultaneous presence, which correspond to the possibility of synchronous and asynchronous work) and reachability (reachable and unreachable actors), it is possible to classify the possible interaction scenarios between two actors (Fig. 4). Provided the collaboration is done on-demand, whenever the actors decide to collaborate they will be in a particular quadrant of the classification. In that case, the actors will need to count on a set of particular services that belong to the back-end of a collaborative application in order to carry out the collaboration process between them.

There are four possible collaboration scenarios: simultaneous and reachable, simultaneous and unreachable, non-simultaneous and reachable and non-simultaneous and unreachable. In the simultaneous and reachable scenarios, both actors are working at the same time and each one is able to interact with the other one directly, e.g., synchronizing collected information. In a simultaneous and unreachable situation, the actors are working synchronously but they are unreachable, and therefore unable to communicate in a predictable way. For example, two users may be at work in the same building on related tasks, but in an extended area in which the communication range of the ad-hoc wireless network is too small to keep them communicated. In the non-simultaneous and reachable scenario, the actors are working in different time periods, but there is an infrastructure (e.g., a computer server) that allows them to communicate asynchronously despite their time differences. Finally, in an asynchronous and non-simultaneous situation, collaboration between two actors is extremely difficult since they are working at different times and lack a way to communicate directly. For example, two actors may work in an extended area at different times. In this case, although work is practically autonomous and weakly interdependent, providing technological support may ease the collaboration process.

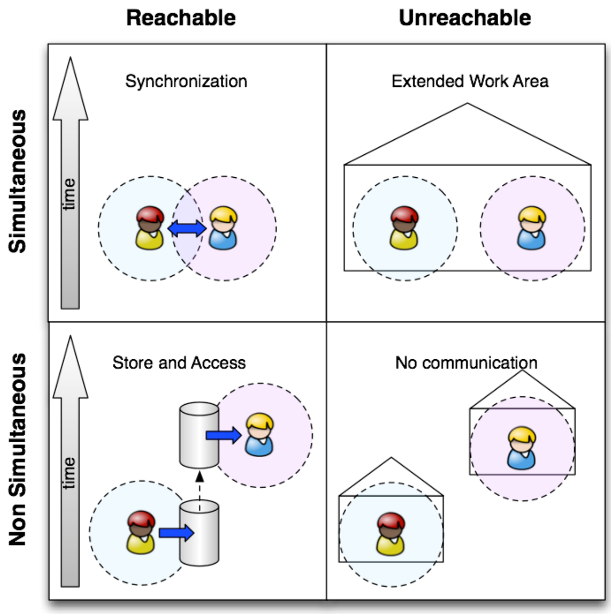


*Figure 4: Classification of Collaboration Scenarios*

The actors' mobility may cause the interaction scenario to change from one quadrant to another, e.g. in the case an actor becomes unreachable due to lack of communication services. In such case, the actors will need to use another set of services (specific to that interaction scenario) in order to continue with the collaboration process. If the collaborative system supporting these actors is context-aware, it can dynamically adapt its functionality in order to continue supporting the collaboration process between these two actors. Otherwise, the mobile collaborative application should implement a set of services, and each user may be in charge of identifying and using the best mechanism to support the collaboration depending on the quadrant characterizing the interaction scenario.

## 4 Requirements for Mobile Collaboration

There are several general (transversal) requirements that have to be considered when developing any software system. Most of them correspond to quality requirements, such as maintainability, flexibility and reliability [Myers 1976]. Some specific requirements have been described for specific types of systems, e.g., for collaborative systems [Mandviwalla and Olfman 1994]. In this section, we propose a framework of general requirements that are usually present in the development of mobile collaborative systems. These requirements were identified based on a study of the limitations and challenges of mobile technology, as well as experiences described in the literature and the authors' experience developing mobile collaborative applications. Each requirement is described below.

**Users Interaction Flexibility** (We will refer to this requirement as ***flexibility***): Mobile collaborative systems must support frequent changes in group size and structure, as mobility may cause group participants to connect or disconnect from the group [Essman and Hampel 2005, Neyem et al. 2007, Pinelle 2004, Pinelle and Gutwin 2005]. A couple of mechanisms to provide flexibility are the following ones:

- *Automatic user detection*: The mobile collaborative workspace has to automatically collect and keep information about the reachable users (peers). In addition, the system has to store the information related to peer availability. Based on that contextual information the collaborative system could implement awareness mechanisms (i.e. user presence, user availability, or user location) that trigger on-demand collaboration processes.
- *User connection / disconnection*: Applications may allow participants to work offline for most of the time and switch to online use on-demand. Thus, participants will be able to choose their own level of involvement in the collaboration according to their needs and situation.

**Users Interaction Protection** (We will refer to this requirement as ***Protection*****):** The collaborative system must incorporate measures ensuring the work of each user is protected [Gutwin and Greenberg 2000, Kortuem et al. 2001, Roth 2002]. Some of these measures are mentioned below:

- *Ad-hoc work sessions*: The interaction among mobile users should be protected in order to avoid unauthorized participation in the group and invalid access to resources shared among them. The transmission process should be secured from external access. Ad hoc work sessions are a mechanism to deal with this need.
- *User privacy*: Each user should be able to choose which data to share, and some actions may be performed privately. Depending on the users heterogeneity in the work scenario, fine-grained authorization management could be required to do an appropriete privacy tuning. Users will be more likely to collaborate if their privacy is respected.
- *Security*: The work of each user must be protected so no one can, maliciously or by mistake, destroy someone else's work. It includes data and communication protection, and also users identity verification.

**Communication** (We will refer to this requirement as ***communication*****):** Mobile collaborative system users need to communicate with each other by exchanging messages (e.g., notes, documents or alarms) [Caporuscio and Invernardi 2003, Gutwin and Greenberg 2000, Neyem et al. 2007, Pinelle 2004]. Since the users are on the move to carry out their activities, some of them could be unreachable during a time period. Therefore, the system should provide mechanisms for synchronous/asynchronous communication and attended/unattended message delivery. Some of these mechanisms are the ones described below:

- *Synchronous messaging*: When two users are simultaneously available and reachable, they should be able to exchange messages, e.g., to synchronize their shared dataspace or receive notifications. This synchronous communication is the base for synchronous collaboration.
- *Asynchronous messaging*: When two users work at different times, the system should permit them to send messages that they will receive when they are available. Examples of asynchronous messaging are electronic mail, or message delivery when a user connects to a server. Like the previous case, asynchronous communication is the base for asynchronous collaboration.
- *File transfer*: Mobile workers carry out weakly interdependent tasks, therefore it is not required that each person should have an instance of each shared resource. Typically members of a particular working group keep, in the local shared workspace, the resources that are relevant to carry out their assigned tasks. New members assigned to that working group could require getting the shared information from their partners in order to start performing a particular task. File transfer is a mechanism required to deal with this need.
- *Pushing notifications*: The messages are delivered to the mobile users at the moment they connect to the server. Typically a pop-up window could be displayed on the user interface to show the pending messages.

**Heterogeneity and Interoperability** (We will refer to this requirement as ***heterogeneity***)**:** Regardless of the mobile device used to access the mobile application, a person should be able to interact with any collaborator using the same application [Caporuscio and Invernardi 2003, Guerrero et al. 2006, Neyem et al. 2008]. Differences among the devices are a burden to the users that should be eased by the collaborative system. Requirements that should be considered are the following ones:

- *Heterogeneity*: Collaboration may involve heterogeneous devices, such as laptops, PDAs and smart-phones. These devices have different hardware features and computing capabilities. It is recommended to design a version of a mobile collaborative application for each device being considered for the collaboration process. This ensures the application will take advantage of the particular device features.
- *Interoperability*: Interoperability refers to the capability of a software system to understand the meaning of data and services, even though these resources have been designed by several providers. Data and services interoperability should be ensured in each version of the system in order to prevent the users from becoming unwillingly isolated.

**Autonomous Interaction-Support Services** (We will refer to this requirement as ***networking***)**:** Many work scenarios do not provide wireless communication support; in those cases the application should work based on a Mobile Ad-hoc Network (MANET) [Caporuscio and Invernardi 2003, Essman and Hampel 2005, Kortuem et al. 2001, Neyem et al. 2007, Pham et al. 2000, Pinelle 2004, Roth 2002, Sacramento et al. 2004]. Networking issues should be transparent for the end-user. Otherwise, the collaboration capability is at risk. The following mechanisms may be used to provide connectivity to users of a mobile collaborative application:

- *Automatic connection*: The MANET should be automatically formed and kept by applications running in each participating node. This increases the interaction capabilities among the participants and it increases the availability of the shared resources.
- *Service and device discovery*: Available services and devices (such as public screens, smartboards, file upload, etc) should be dynamically detected and seamlessly integrated into the collaborative environment surrounding the physical work context. Some mechanisms for this have been designed especially for MANET [Campo et al. 2006]. This may support the collaboration process during casual interactions.
- *Message routing*: This communication mechanism uses intermediary mobile workers to provide reachability between two actors that have more than a one-hop distance between them. Message routing transforms a one-hop network (with a limited communication threshold) into a multi-hop network (with a larger communication threshold).
- *User gossip*: This service is based on gossip sent by a user (interested in starting a collaboration process) to the unreachable partner through the intermediary neighbor nodes. The movement of such nodes (and the unreachable partner) eventually allows message delivery. Typically the message carries information about the requester's locations in the near future. That information (if it is received by the unreachable partner) could ease communication between them.

**Users Awareness** (We will refer to this requirement as ***awareness***)**:** Since a mobile collaborative system supports group work, it must provide awareness mechanisms to improve the users' understanding of each others' work [Gutwin and Greenberg 1999, Gutwin and Greenberg 2000, Pinelle 2004, Sacramento et al. 2004]. The system should offer offline and online awareness, both of which should be updated as information becomes available. This information should be presented but should not overwhelm the recipients, since this may diminish their ability to perceive it [Papadopoulos 2006]. The types of awareness that should be supported are at least the following:

- *Online awareness*: Examples of online awareness are lists of connected users, user locations, and current activity.
- *Offline awareness*: Examples of offline awareness are last available modification to a document, and text authorship.

- *Transition awareness*: Awareness about the transitions between connection and disconnection, such as user presence awareness with time memory, or awareness of message delivery.

**Data Consistency and Availability** (We will refer to this requirement as ***information support***)**:** Frequent disconnections and autonomous work usually cause inconsistencies and unavailability of the resources that are being shared by the group members [Essman and Hampel 2005, Neyem et al. 2008, Roth 2002]. Some requirements supporting consistency and availability of data are the following ones:

- *Explicit data replication*: During connection periods, a user should be able to share data with another user so this data will be available to both of them when they are no longer reachable.
- *Caching*: When users collaborate, as much of the shared data as possible may be automatically replicated in each user's workspace, in order to provide each user with the most up-to-date information when doing subsequent autonomous work.
- *Conflict resolution*: Mobile workers may update local information on the mobile collaborative application when working alone. Eventually, this may generate inconsistencies in the shared data. Data synchronization requires conflict resolution algorithms to reconcile that information in order to have a common view of the shared environment [Lukosch 2008].

### 4.1 Correspondence Matrix

This section presents a correspondence matrix that illustrates how each general requirement affects the other requirements (Table 1). The impact may be positive (if it contributes to the other's accomplishment), negative (the contrary case) or neutral (both requirements are independent).

Table 1 shows flexibility, or the fact that the group may become disconnected at any moment, negatively impacts communication, networking, awareness and information support. The cause for this association is that frequent disconnections - loosely coupled work -decrease group cohesion and the possibilities for direct or indirect communication.

Information support positively impacts awareness, since shared information is replicated every time users communicate. At any moment, replicated information may be used for offline awareness of other users' work.

Heterogeneity is negatively related to communication because the more heterogeneous a group is in term of devices and software, the more difficult it is to build interoperable mobile collaborative systems to support the group's communications. Finally, an increase in awareness information provided to users causes their privacy to decrease.

| | Protection | Communication | Heterogeneity | Networking | Awareness | Information Support |
|---|---|---|---|---|---|---|
| Flexibility | | - | | - | - | - |
| Protection | | - | | | - | |
| Communication | | | - | | | |
| Heterogeneity | | | | | | |
| Networking | | | | | | |
| Awareness | | | | | | + |

+ : $x_i$ positively impacts $x_j$
- : $x_i$ negatively impacts $x_j$

*Table 1: Relating general requirements*

### 4.2 Grouping General Requirements

Requirements may be grouped as they often appear in mobile collaborative application development. Some requirements are tightly related to others, so that when one requirement is identified for a development project, it is highly probable that its related requirements are also present.

*Flexibility, Information support*: When users work autonomously, flexibility is maximized, since users are usually disconnected and they only connect for short periods of time. In this case, periods of connectivity must be used to replicate as much information as possible, to be available to users later on when they are disconnected. This combination of requirements is usual in loosely coupled group work.

*Networking, Awareness, Protection*: The automatic detection and configuration of a network of users and devices requires awareness mechanisms to ensure users understand who is in the network and how they might interact. Automatic networking services also require protection from malicious users trying to connect to the network and privacy mechanisms in case, e.g., a user does not want to be contacted.

Requirements may also be grouped according to the collaboration scenario quadrants in which they are required. Since collaboration may transition from one quadrant to an adjacent one, requirements may also be needed to ease the transitions. This grouping permits developers to know which requirements they should consider during system development according to how collaboration will take place. Each requirement may be present in one or more collaboration scenarios or transitions. The following requirements: service and device discovery, heterogeneity, interoperability, user privacy, and security have not been added because they are not related to a collaboration situation between two users.

The most intensive collaboration scenario occurs when two users are working synchronously and are reachable. This situation requires them to be able to work together collaboratively by exchanging messages, files and data, as well as providing services that improve collaboration such as awareness and automatic networking mechanisms. On the opposite extreme is the non-simultaneous and unreachable situation, in which collaboration between two users cannot take place because there is no way for them to interact. In this case, the only suggested requirement is to provide offline awareness by displaying awareness information received during connection periods.

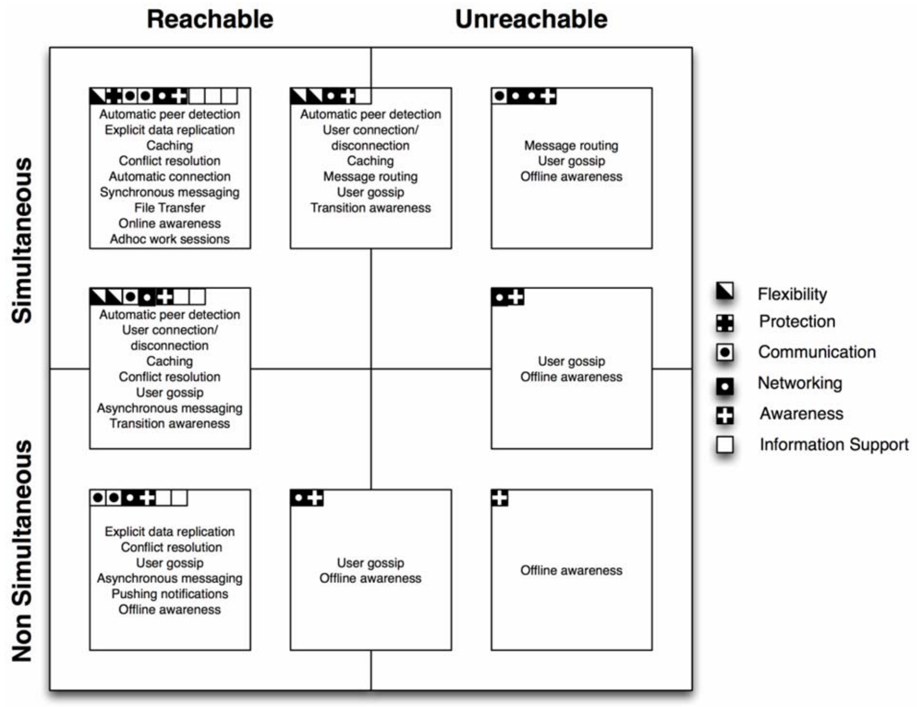


*Figure 5: Requirements organized according to the collaboration classification*

Figure 5 presents the requirements that are present in each of the collaboration scenarios. Each quadrant also displays a colored square that represents the category each requirement belongs to. From this figure, we may observe, e.g., that awareness must always be present and that efforts should be made to connect two users even if they are neither reachable nor working simultaneously.

## 5 Experiences Developing Mobile Collaborative Applications

This section presents two recently developed mobile shared workspaces (MSW) to illustrate how the proposed requirements apply to each case. A MSW is a persistent space in which users interact through their mobile devices [Schaffers et al. 2006]. The first one is named MobileMap, and it was designed to support Chilean firefighters doing search and rescue activities during emergency situations [Ochoa et al. 2007, Monares et al. 2011]. The second one is a MSW to support construction personnel inspecting the physical infrastructure in building projects [Ochoa et al. 2011].

### 5.1 MobileMap

A scenario for mobile collaborative technologies application is the case of firefighters responding to emergency situations such as earthquakes and floods. This scenario includes firefighters constantly moving in order to assess the affected area and possible hazardous events they must deal with. In these situations, firefighters generally communicate face-to-face and through radio to exchange information on resource allocation, location of injured or trapped people, dangerous areas and roles of the surrounding buildings (e.g. hospitals, schools or homes). Typically, only one radio channel is available to carry out the response activities. During these emergencies the existent communication infrastructure in the affected area (e.g. wired telephony and cellular network) is usually unavailable, inexistent or collapsed.

However, firefighters can take advantage of wireless communication and mobile computing to support their work; particularly the coordination and collaboration processes among them. In order to allow collaboration in emergency situations, a MSW called *MobileMap* was developed to support firefighters' work [Ochoa et al. 2007, Monares et al. 2011]. The software was implemented in C# for Windows Mobile 5.0 and it was tested in several PDAs embedding a GPS device.

MobileMap is used by the leader of a team, who is in charge of coordinating firefighters within his team and also with external collaborators (e.g. other group leaders and the emergency managers). The infrastructure supporting the communication among them is one or more Mobile Ad-hoc Networks (MANET) [De Rosa et al. 2005] linking the mobile devices deployed in the affected area. MobileMap displays a street map that includes the team leader's current location, the affected area, and the points of interest (e.g. hospitals, schools, fire stations, or police offices). The users can mark the map with additional points of interest, e.g. location of trapped victims, evacuation routes or places to assist injured persons. The software also permits team leaders to share and synchronize the information they have in their PDAs through a MANET (Figure 6).


*Figure 6: Firefighters sharing points of interest through MobileMap*

The work of firefighters during search and rescue operations in an emergency situation can be characterized as loosely coupled, since they are spread over a large and irregular area. Each firefighter works on his PDA. The MSW is fully-replicated in terms of services and data. Next, we show how the general requirements presented in section 4 were considered in the design of this MSW.

- *Flexibility*. Firefighters usually work in a large area doing loosely coupled work, thus most of the time their PDAs are disconnected. However, they may choose to synchronize their information at any moment. The MSW manages the situations where firemen are synchronizing information and become disconnected.
- *Protection*. Firefighters work with sensitive information. However, they work publicly and usually over radio channels that may be easily overheard, so privacy is not relevant in the case of the firemen's work. Thus, no additional functionality to deal with this requirement was included in the current version of the MSW.
- *Communication*. Firefighters choose when and which information to synchronize with other firefighters. The mechanism used for message delivery can act as an attended or unattended process. This mechanism uses message routing.
- *Heterogeneity*. The MSW was implemented using functionality available in the .Net Compact Framework. Therefore, the application is able to run on a large range of devices: from smartphones to desktop PCs. The supporting framework provided the service interoperability.
- *Networking*. The MSW automatically form a MANET composed by all reachable mobile computing devices. The connection/re-connection of mobile units to the network is automatic and transparent for end-users.
- *Awareness*. The current version of the software does not include awareness mechanisms in the MSW. However, each mobile unit keeps an updated list of reachable users; hence online connection awareness could be easily included.
- *Information support.* The coherence of the shared information is kept with on-demand replication and synchronization commands. The synchronization process may be attended or unattended.

In this case, we may see the group of requirements that is strongly present is the group containing flexibility and information support. Firefighters work in loosely coupled teams; this implies they are generally working autonomously. Their work is usually urgent, so disconnections are frequent and flexibility is highly important. Furthermore, the physical areas where they work may overlap and they will need to coordinate and synchronize their information very quickly, making sure information is consistent to avoid duplicate efforts and taking unnecessary risks. We also may see that firefighters work in all of the collaboration scenarios represented in Fig. 5, so every listed requirement may be applied to this collaborative system.

MobileMap was first evaluated by two experts from the 6$^{th}$ and 8$^{th}$ Santiago (Chile) firefighter units [Monares et al. 2011]. Then, the application was evaluated by 3 firefighters from these companies. In all cases the MSW was used running on PDAs. The users evaluated the system by simulating the actions that firefighters must do during two small urban emergency situations: a home fire and a car accident. The characteristics of these emergencies were obtained from real cases that were occurring in the city. The details of those events were received through the radio system located at the Santiago Alarms Center.

After the experience a questionnaire was given to the firefighters in order to evaluate MobileMap. The questionnaire evaluated the MSW functionality, performance and usability in terms of the requirements presented above. The first important conclusion indicates the system is ready to use at least in small urban incidents (fires, chemical spills, and small collapses). The system functionality was considered useful to support urban search and rescue activities. The services embedded in the application to support collaboration worked well; therefore the coordination and sharing information processes were easy to do. The users evaluated the application usability and performance as good, and the functionality as very good.

### 5.2 MSW for Construction Inspections

Another scenario where mobile collaboration is present is in construction inspections. Typically, each construction site has a main contractor. The main contractor in turn outsources several parts of the construction project, e.g. electrical facilities, gas/water/communication networks, painting and architecture. Some of these sub-contracted companies work concurrently and they have to collaborate in order to know each other's progress to plan the execution of their own pending work. Moreover, all these companies should periodically report their progress to the main contractor; the contractor must coordinate the efforts of the companies. For example, electrical engineers working for an electric sub-contractor company need to be on the move in order to inspect and record the state of the electrical facilities being developed by company workers at a construction site. During the inspection, each engineer using a Tablet PC updates the information recording the current state of the electrical facilities (Figure 7).

*Figure 7: Use of mobile technology in construction scenarios*

After the inspection and before leaving the construction site, the electrical engineers meet to share the data, review it and check agreements on the updated information. If they detect incomplete or contradictory data, some of them can re-inspect the facilities in order to solve such cases. Before leaving the construction site, an electrical engineer shares the updated information with the employee who is in charge of tracking the construction project updates for the main contractor.

A MSW was developed to carry out these inspection activities [Ochoa et al. 2011]. The application allows users to manage the blueprint of the construction project, to do annotations on the maps, and to synchronize and share the annotations/maps using any type of wireless network.

The MSW is fully distributed and runs on Laptops and TabletPCs; however, the user interface was designed to be used on a TabletPC. Next, we analyze how the general requirements presented in section 4 influenced the design of this MSW.

- *Flexibility*. Construction inspectors carry out loosely coupled work most of the time, and they interact with others (partners, representatives from the main contractor, other participating professionals) on-demand. The MSW creates and automatically keeps a MANET, so the users become visible to each other when they are inside the communication range. The MSW allows users to choose the working mode from two options (available or busy) in order to avoid unwanted disruptions. “Available” means the users can interact with each other. Otherwise, the working mode will be “busy”.
- *Protection*. Mobile workers are organized in sessions to avoid unauthorized access to shared resources. A user can belong to more that one session. If two users belonging to different sessions want to collaborate, they can create a session for themselves or they can interact in the public space (out of any session).
- *Communication*. The communication infrastructure used by the MSW is also the same as explained for the previous case.
- *Heterogeneity*. The MSW has only one version, used for TabletPC, Laptop and Desktop PC. The solution is interoperable when it runs in such devices. Currently, a lightweight version of this MSW is being developed for PDAs. Both versions should be able to interoperate between them.
- *Networking*. This functionality is the same as explained for the previous work scenario.
- *Awareness*. The MSW provides user connection awareness, and also awareness of the resources that are shared in each work session the user belongs to. In addition, the application provides awareness about the ownership of annotations shown on a map.
- *Information support.* The components used to keep the information coherence are those explained for the previous work scenario; therefore the functionality in both cases is the same.

In this case, we may see that both groups of requirements are present. First, flexibility and information support must be present to avoid work duplication and data that is not up-to-date. Also, networking, awareness and protection are most important in this case because users may be working in an extended area and opportunistic collaboration should be encouraged through awareness mechanisms and strategies for best connectivity. In this case, data (such as the result of an inspection) may be sensitive and should be protected from malicious users.

A preliminary evaluation of the application was done by two civil engineers who used to be involved in construction inspection processes [Ochoa et al. 2011]. They ran the application on a TabletPC. Similar to the previous case the engineers simulated the activities they would have to do to inspect physical facilities. For that purpose, the authors provided a project (with the corresponding maps) to the inspectors. Once they opened the project, they made annotations, created/modify tasks and synchronized the project between them.

After the experience they filled the same questionnaire mentioned for the previous experience. They found the performance and usability of the tool was good, and the functionality highly useful. Besides, they thought the time and cost of the coordination processes could be reduced to very low values when using the MSW. However, they highlighted that users must feel comfortable using the application and they must trust it. Otherwise, the possible benefits will not be obtained. Two additional aspects they highlighted were related to the use of TabletPCs as mobile devices. They thought a battery providing additional use time was needed. They also disliked the need to charge these devices during the inspection processes.

Based on the evaluation results of these two MSW, it seems that considering the proposed general requirements as an input for the design of mobile collaborative applications could have a positive impact. However, the analysis of two applications is not enough to get strong conclusions about that. To address this shortcoming, the authors have just finished the implementation of a MSW to support undergraduate/graduate students in their instructional activities (e.g. work meetings, exams, annotations taking and group work). The application runs on a TabletPC, and the next step is to carry out an evaluation process similar to the ones described above.

## 6 Conclusions and Future Work

Developing mobile applications to support collaborative work is a challenging task, since this is a recent area and software must overcome challenges in technical implementation, collaboration support and users' adoption.

The requirements proposed in this paper are based on extensive work done by many researchers and our own experience developing mobile collaborative systems. Beginning development with such a list would greatly ease building the software, especially for developers without experience in this area, since these requirements are usually hidden and they should be considered in each development. Naturally, these requirements are a fraction of all the mobile collaboration back-end requirements. Other requirements should also be considered, e.g., requirements to support the autonomous work that users do in periods of isolated work.

Ignoring the presented requirements may cause developers to build a system from scratch that is missing a fundamental component that supports or eases collaboration. Naturally, some requirements may be intentionally ignored according to the application's particular requirements. For instance, a mobile application whose goal is to have as many users as possible reading and commenting news articles will not need to implement ad-hoc work sessions to protect shared data.

Classifying mobile collaborative system requirements according to the possible collaboration scenarios allows us to realize that not all mobile collaboration takes place in the same way. Therefore, the requirements that should be present depend on the types of interactions users need. A challenging moment for collaboration is when users are asynchronous and unreachable, and further study is required to improve collaboration support in this case or to help avoid these situations.

Future work will include further tests of the MSW applications to refine and improve the software and our understanding of mobile collaboration requirements. For the same purpose, the requirements will be applied to other developments of mobile collaborative applications. The proposed requirements may also be used to build a development framework for mobile collaborative systems, as well as an evaluation method to indicate whether requirements are successfully met.

**Acknowledgements**

This work was partially supported by Fondecyt (Chile), grants Nº: 11060467, 1080352 and 11090224.

## References

[Campo et al. 2006] Campo, C., Almenárez, F., Díaz, D., García-Rubio, C., López, A.M.: “Secure Service Discovery based on Trust Management for Ad-hoc Networks”; J.UCS (Journal of Universal Computer Science), 12, 3 (2006), 340-356.

[Caporuscio and Invernardi 2003] Caporuscio, M., Invernardi, P.: “Yet another framework for supporting mobile and collaborative work”; Proceedings of WETICE '03: Twelfth International Workshop on Enabling Technologies, IEEE CS Press, Washington DC (2003), 81.

[Carter et al. 2004] Carter, J. Mankoff, P. Goddi, P.: “Building Connections among Loosely Coupled Groups: Hebb’s Rule at Work”; Computer Supported Cooperative Work Journal, 13, 3-4 (2004), 305-327.

[Churchill and Wakeford 2001] Churchill, E. and Wakeford, N.: “Framing mobile collaborations and mobile technologies”; In B. Brown, N. Green, R. Harper (eds): Wireless world: social and interactional aspects of the mobile age, Springer-Verlag, New York (2001), 154-179.

[Cheverst 1999] Cheverst, K.: “Development of a Group Service to Support Collaborative Mobile Groupware”; PhD Thesis, Lancaster University (1999).

[De Rosa et al. 2005] De Rosa, F., Malizia, A., Mecella, M.: "Disconnection Prediction in Mobile Ad hoc Networks for Supporting Cooperative Work"; IEEE Pervasive Computing, 4, 3 (2005), 62-70.

[Divitini et al. 2004] Divitini, M., Farshchian, B., Samset, H.: “UbiCollab: Collaboration support for mobile users”; Proceedings of SAC '04: ACM Symposium on Applied Computing, ACM Press, New York (2004), 1191-1195.

[Dix et al. 1998] Dix, A., Finlay, J., Abowd, G., Beale, R.: “Human-Computer Interaction”, Prentice Hall (1998)

[Ellis et al. 1991] Ellis, C.A., Gibbs, S.J., Rein, G.: “Groupware: some issues and experiences”; Communications of the ACM, 34, 1 (1991), 39-58.

[Essman and Hampel 2005] Essman, B., Hampel, T.: “A design pattern for mobile-distributed knowledge spaces”; Proceedings of MIS '05: Symposia on Metainformatics, ACM Press, New York (2005), 3.

[Guerrero et al. 2006] Guerrero, L., Ochoa, S., Pino, J.A., Collazos, C.: “Selecting Devices to Support Mobile Collaboration”; Group Decision and Negotiation, 15, 3 (2006), 243-271.

[Grudin 1994] Grudin, J.: “CSCW: History and Focus”; Computer, 27, 5 (1994), 9-26.

[Gutwin and Greenberg 1999] Gutwin, C., Greenberg, S.: “A Framework of Awareness for Small Groups in Shared-Workspace Groupware”; Technical Report 99-1, Department of Computer Science, University of Saskatchewan, Canada. (1999).

[Gutwin and Greenberg 2000] Gutwin, C., Greenberg, S.: “The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces”; Proceedings of WETICE '00, IEEE CS Press, Washington DC (2000), 98–103.

[Kortuem et al. 2001] Kortuem, G., Schneider, J., Preuitt, D., Thompson, T. G., Fickas, S., Segall, Z.: “When Peer-to-Peer comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Ad hoc Networks”; Proceedings of P2P '01: First International Conference on Peer-to-Peer Computing, IEEE CS Press, Washington DC (2001), 75.

[Lukosch 2008] Lukosch, S.: “Seamless Transition between Connected and Disconnected Collaborative Interaction”; J.UCS (Journal of Universal Comp. Science), 14, 1 (2008), 59-87.

[Mandviwalla and Olfman 1994] Mandviwalla, M. and Olfman, L.: “What do groups need? A proposed set of generic groupware requirements”; ACM Transactions on Computer-Human Interaction, 1, 3 (1994), 245-268.

[Menchaca-Mendez et al. 2004] Menchaca-Mendez, R., Gutierrez-Arias, E., Favela, J.: “Opportunistic Interaction in P2P Ubiquitous Environments”; Lecture Notes in Computer Science 3198 (2004), 349-362.

[Monares et al. 2011] Monares, A. Ochoa, S. F., Pino, J. A., Herskovic, V., Rodriguez-Covili, J., Neyem, A. : “Mobile Computing in Urban Emergency Situations: Improving the Support to Firefighters in the Field”; Expert Systems with Applications, 38, 2 (2011), 1255-1267.

[Munson and Dewan 1997] Munson, J. P. and Dewan, P.: “Sync: A Java Framework for Mobile Collaborative Applications”; Computer, 30, 6 (1997), 59-66.

[Myers 1976] Myers, G.: “Software Reliability, Principles and Practices”; Wiley, New York (1976)

[Neyem et al. 2007] Neyem, A., Ochoa, S., Pino, J.: “Designing Mobile Shared Workspaces for Loosely Coupled Workgroups”; Lecture Notes in Computer Science 4715 (2007), 173-190.

[Neyem et al. 2008] Neyem, A., Ochoa, S.F., Pino, J.A.: “Integrating Service-Oriented Mobile Units to Support Collaboration in Ad-hoc Scenarios”; J.UCS (Journal of Universal Computer Science), 14, 1 (2008), 88-122.

[Ochoa et al. 2007] Ochoa, S.F., Neyem, A., Pino, J., Borges, M.: “Supporting Group Decision Making and Coordination in Urban Disasters Relief Efforts”; Journal of Decision Systems 16, 2 (2007), 143-172.

[Ochoa et al. 2011] Ochoa, S.F., Bravo, G., Pino, J., Rodriguez, J.F.: “Coordinating Loosely-Coupled Work in Construction Inspection Activities”. Group Decision and Negotiation 20, 1 (2011), 39-56.

[Papadopoulos 2006] Papadopoulos, C.: "Improving Awareness in Mobile CSCW", IEEE Transactions on Mobile Computing, 5, 10 (2006), 1331-1346.

[Parsons et al. 2007] Parsons, D., Ryu, H., Cranshaw, M.: "A Design Requirements Framework for Mobile Learning Environments", Journal of Computers, 2, 4 (2007), 1-8.

[Pham et al. 2000] Pham, T., Schneider, G., and Goose, S.: "A situated computing framework for mobile and ubiquitous multimedia access using small screen and composite devices"; Proceedings of MULTIMEDIA '00: Eighth ACM International Conference on Multimedia, ACM Press, New York (2000), 323-331.

[Pinelle 2004] Pinelle, D.: "Improving Groupware Design for Loosely Coupled Groups"; PhD Thesis, University of Saskatchewan (2004).

[Pinelle and Gutwin 2005] Pinelle, D., Gutwin, C.: "A Groupware Design Framework for Loosely Coupled Workgroups", Proceedings of ECSCW '05: European Conference on Computer-Supported Cooperative Work, Springer-Verlag, New York (2005), 119-139.

[Pinelle and Gutwin 2006] Pinelle, D., Gutwin, C.: "Loose Coupling and Healthcare Organizations: Deployment Strategies for Groupware"; Computer Supported Cooperative Work Journal, 15, 5-6 (2006), 537-572.

[Roth 2002] Roth, J.: "Seven Challenges for Developers of Mobile Groupware"; CHI '02 Workshop on Mobile Ad Hoc Collaboration (2002).

[Sacramento et al. 2004] Sacramento, V., Endler, M., Rubinsztejn, H. K., Lima, L. d., Goncalves, K., and Bueno, G. A.: "An Architecture Supporting the Development of Collaborative Applications for Mobile Users"; Proceedings of WETICE '04, IEEE CS Press, Washington DC (2004), 109-114.

[Schaffers et al. 2006] Schaffers, H., Brodt, T., Pallot, M., Prinz, W. (eds.): "The Future Workplace - Perspectives on Mobile and Collaborative Working", Telematica Instituut (Enschede), The Netherlands (2006).

[Tietze 2001] Tietze, D.: "A Framework for Developing Component-based Co-operative Applications" PhD Thesis, Technischen Universität Darmstadt (2001).

[Zurita et al. 2007] Zurita, G., Antunes, P., Baloian, N., Baytelman, F.: "Mobile SenseMaking: exploring Proximity and Mobile Applications in the Classroom"; J.UCS (Journal of Universal Computer Science), 13, 10, (2007), 1434-1448.

[Zurita et al. 2008] Zurita, G., Antunes, P., Baloian, N., Carrico, L., Baytelman, F., de Sá, M.: "Using PDAs in Meetings: Patterns, Architecture and Components"; J.UCS (Journal of Universal Computer Science), 14, 1, (2008), 123-147.