Identifying the Awareness Mechanisms for Mobile Collaborative Applications

Valeria Herskovic[1], Sergio F. Ochoa[2], José A. Pino[2], Pedro Antunes[3], and Emilio Ormeño[4]

[1] Computer Science Department, Pontificia Universidad Católica de Chile
Av. Vicuña Mackenna 4860, Macul, Santiago, Chile
vherskov@ing.puc.cl

[2] Department of Computer Science, Universidad de Chile
Av. Blanco Encalada 2120, 3rd Floor, Santiago, Chile
{sochoa,jpino}@dcc.uchile.cl

[3] School of Information Management, Victoria University of Wellington
Wellington, New Zealand
pedro.antunes@vuw.ac.nz

[4] Institute of Informatics, Universidad Nacional de San Juan
San Juan, Argentina
eormeno@idei.unsj.edu.ar

Abstract. The complexity of modeling collaborative systems has been broadly recognized by the CSCW community. Mobile collaborative applications are a particular case of those systems, where design requirements and constraints are even more complex than in stationary solutions. Design complexity in mobile application increases because mobility changes the interaction requirements of nomadic users and the capabilities of devices to support them. Consequently, the awareness support provided by these systems should also be adjusted according to the nomadic users' context. This article presents a method that helps identifying the awareness mechanisms required by nomadic users to support a certain activity. The method, named Awareness Identification Method for Mobile Applications (AIMMA), suggests particular awareness components embedded in mobile collaborative applications, which will increase the interaction possibilities of users participating in a collaborative process. AIMMA can be used by software developers as a design guideline. This article reports the results of a proof of concept where the proposed method helped identifying suitable awareness mechanisms to improve the collaboration support of a mobile application. This method could also be extended to help identify, e.g., the services required by mobile workers to support their interactions.

Keywords: Mobile collaboration, awareness mechanisms, software design, users interaction, system evaluation.

1 Introduction

The complexity of modeling collaborative systems has been broadly recognized by the CSCW community. The success of a collaborative system depends on multiple factors,

P. Antunes et al. (Eds.): CRIWG 2013, LNCS 8224, pp. 241–256, 2013.

such as the group's characteristics, the work context, and the effects of technology on the supported collaborative activity [1]. Besides modeling, the design of these systems is also complex [14]. One reason is that most developers have experience implementing single-user applications, but little experience with multi-user applications. This single-user bias has been shown to affect the developers' intuition on what makes an application successful [10, 11]. When the system under development is mobile, design complexity increases because users' mobility changes the physical, task and interaction contexts. In particular, this means that a mobile application should dynamically self-configure its services to the new conditions and constraints.

Frequently, mobile collaboration happens in a loosely coupled way. This suggests that groups have low interdependence, high differentiation, low integration, and that these characteristics remain stable over time, resulting in groups that work autonomously and weakly depend on each other [24]. As a consequence, multiuser interaction is sporadic, occurring only when users require it. Collaborative systems should ease multiuser interaction at these moments, providing awareness mechanisms to understand the collaborators' activities and whereabouts, while promoting participation and collaboration.

Unfortunately there are few guidelines to identify which interaction services should be made available to increase coupling under certain adverse conditions, and also what awareness mechanisms could trigger such interactions. Herskovic et al. state that the selection of multiuser interaction services and awareness mechanisms depend on the supported task and also the interaction context [13]. The task dependency cannot be addressed in a transversal way (e.g. using a "fixed" set of Questions, Options and Criteria), because each task has its own particularities. However the second aspect, i.e. interaction dependency, can be addressed as a transversal design issue. Let us illustrate this statement. Assume for simplicity that only two mobile users are collaborating mediated by their communication devices. Their interaction context may be one of four possibilities, which depend on 1) the simultaneity of the presence of the two users at the moment they decide to interact, and 2) the capability of a user to reach the other user [14]. This classification opens up the opportunity to develop specific suggestions about what awareness mechanisms to support in each scenario. Note that these suggestions must be dynamic because people on the move can change their connectivity, which has implications on the interaction context.

Trying to contribute to reduce the design complexity of collaborative systems, this article proposes an Awareness Identification Method for Mobile Applications (AIMMA). The method highlights what awareness mechanisms developers should consider when tailoring technology support to mobile collaborative activities. AIMMA can be used both in the systems development and task/process reengineering cycles, and may even contribute to bring these two important tasks together. This method can be used, with minimal adaptations, to identify other services that must be embedded in a mobile collaborative application; for instance, services to support users' interactions and data sharing.

Next section discusses the challenges posed by awareness support in mobile collaborative applications. That section also presents some related works. Section 3 describes the proposed method. Section 4 presents the tool that supports the AIMMA method. Section 5 describes a case study where AIMMA was used and discusses the obtained results. Section 6 presents some conclusions and further work.

2 Identifying Awareness Mechanisms for Mobile Applications

Identifying which multiuser interaction and awareness mechanisms should be embedded in a mobile collaborative application is a difficult task. According to Herskovic et al., the difficulty in making these choices is a consequence of the iceberg effect [14], i.e. lack of visibility of groupware features, especially for designers and developers with little experience in collaborative systems.

The systems' functional requirements (i.e. those that are focused on single-user interactions and have a representation on the application user-interface) usually tend to be clearly visible for users and developers. They represent the visible part of the iceberg (Fig. 1). On the other hand, groupware requirements (i.e. those involving multiuser interactions) are often known by users but not clearly visible for most developers. One reason is that collaboration support is absent in most common systems, which results in a lack of familiarity for regular software developers. Another reason to ponder is that users often just tacitly know how they collaborate and may find it difficult to describe all details involved in their collaborations. As a consequence, the elaboration of groupware requirements may have to involve people with some experience in the design of collaborative tools.

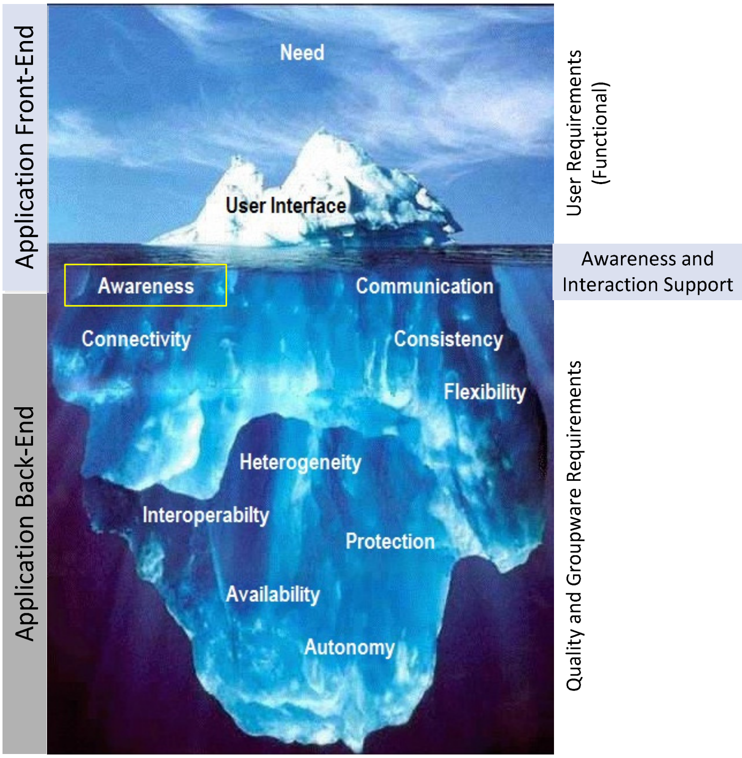

Fig. 1. Representation of the Iceberg Effect, from [14]

In the lower part of the iceberg we include a set of non-functional requirements that often impact collaboration support but may only be identified almost exclusively by groupware experts. The proposed method is focused on identifying those awareness mechanisms that are potentially useful to support multiusers interactions in mobile collaborative applications. This is a quite new research area, in which several initiatives are currently under way.

The research by Edwards et al. [4] concerned asynchronous work by a group of collaborators. They developed a platform called Bayou, on top of which collaborative applications can be built. Application developers can use Bayou to describe the semantic constraints of their applications. Developers can define data-integrity constraints, conflict detection and resolution procedures, and data propagation policies. Izadi et al. [17] propose their own middleware—called FUSE—to help developing mobile collaborative applications. FUSE provides a number of pre-packaged awareness widgets for gathering, distributing and presenting context information to application instances and their users. Each widget manages a set of context information, which helps individuals identifying other group members and coordinating their tasks.

When transforming single-user into multiuser applications, workspace awareness can be obtained as a result of Transparent Adaptations (TA), which are based on the Operational Transformation (OT) technique [28]. In the case of collaborative web-based applications, Heinrich et al. proposed a generic infrastructure promoting the accelerated, cost-efficient development of awareness widgets, as well as non-invasive integration of awareness support with existing web applications [12].

A very interesting paper by Oulasvirta et al. [23] addresses the provision of awareness in a mobile collaborative environment. They assert that "*instead of basing design on trial and error, we aim to reduce uncertainty in design choices by grounding them on findings in social psychology… Our starting point is that the usefulness of a situation cue in inferring another party's current situation depends on two processes: 1) on the individual's correct inference of a situation cue, and 2) on the social interaction afforded by that situation cue*". With that in mind, they developed 11 design requirements based on an equal number of social interaction scenarios. They used those requirements to build *ContextContacts*, a contact book that provides cues about the current situation of other users.

3 Awareness Identification Method for Mobile Applications

The AIMMA method was designed to help developers identify awareness mechanisms that could be useful for supporting mobile collaboration, depending on the users' interaction context at the time they decide to collaborate. Thus, this proposal contributes to reduce the systems' design complexity. AIMMA has three main steps: (1) modeling the multiuser interactions supported by a collaborative mobile application; (2) identifying the list of awareness components which are likely to be included in the application; and (3) reviewing the application to check whether it already includes the suggested awareness mechanisms, and if not, suggesting their integration in the application. Next sections explain these steps in detail.

3.1 Modeling Multiuser Interactions

During this step the designers use the Mobile Collaboration Modeling (MCM) language [13] to build a MCM graph. This graph identifies the roles of users participating in the collaboration process, describes the relationships among these roles, and determines the multiuser interaction contexts that are present in the mobile collaboration scenario. This modeling task must be done by observing and interviewing users.

The specification of this model allows developers to evaluate the completeness and correctness of the interaction scenarios that should be supported by a mobile application. This ensures that the awareness mechanisms to be analyzed are those potentially relevant according to the considered interaction scenarios. The visual representation of this model contributes to reduce the effort required for its evaluation.

Figure 2 shows an example of an MCM graph that describes the roles participating in a collaborative classroom activity, such as a teacher assigning students to teams, defining a leader, sending teams to collect plant specimens, collaborating to build a report, and having the group leader present some conclusions to the teacher. The MCM graph displays roles and how they interact by showing the types of multiuser interaction contexts in which the users can be when they decide to interact. The black squares characterize the arcs between nodes (roles), specifying which multiuser interaction contexts must be supported by the mobile collaborative application, as explained below.

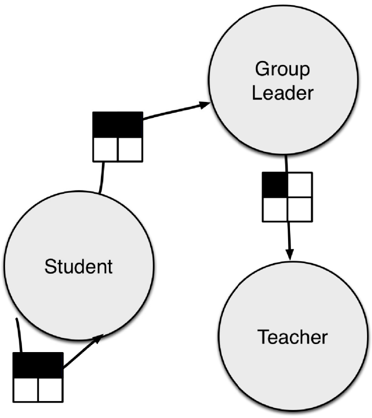

Fig. 2. MCM interaction graph among participants in a collaborative classroom activity

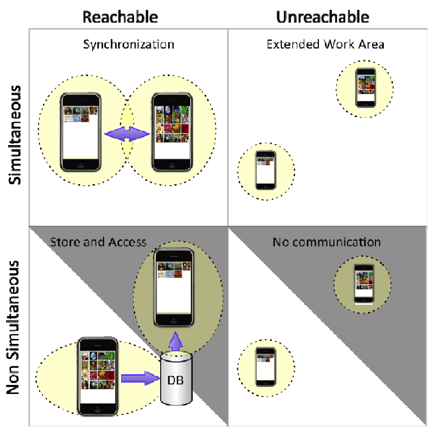

Fig. 3. Multiuser interaction context between two mobile users

The MCM graph depicted in Fig. 2 shows that users play one of the following three roles (light gray nodes): teacher, student, and group leader. In the case shown, interactions can be established when users are in one of the following quadrants: "reachable-simultaneous", and "unreachable-simultaneous". Fig. 3 presents the general multiuser interaction context in the four quadrants of this taxonomy. Note that the labels on the arcs in the MCM graph (Fig. 2) correspond to the interactions shown in Fig. 3.

According to the taxonomy, whenever a user decides to interact with another, that makes them *reachable* if there is an available communication channel between them. In other cases we say that both users are *unreachable*. For instance, two users are unreachable if one of them is unavailable to collaborate or disconnected. Moreover, considering the simultaneity dimension, we can say that two users are *simultaneous* if they are present in a virtual or physical space at the same time. In other cases, we say that both users are non-simultaneous. This is an updated version of the classical space/time CSCW matrix [2], since the ubiquity of mobile devices available today allows users to continuously move between places, making their simultaneity to do work a more relevant distinction. Typically, non-simultaneity occurs when the users collaborate in different shifts. Multiuser interactions between roles can change from one quadrant to another, e.g., because of users' mobility, network access, or changes in their availability. It is important to note that unreachability and non-simultaneity are different: reachability refers to an accessible communication channel and availability to work, which can happen when the users are simultaneous (e.g. face-to-face) or non-simultaneous (e.g. e-mail).

Knowing which roles are participating in multiuser interactions, the relationships among them, and the quadrants in which those interactions take place, allow us to move to the second step, as explained below.

3.2 Identifying Candidate Awareness Components

The second step attempts to identify what types of awareness support are required in each multiuser interaction quadrant. Awareness support is mostly needed when users are working autonomously and at some instant in time they need to interact. Such interactions are typically eased if the application provides awareness about the other users' location, activities, and communication possibilities.

Table 1 summarizes the types of awareness considered by the proposed method, which were obtained from literature and product reviews, using a previous study as a starting point [15]. For each type of awareness we considered the *time* dimension as a transversal factor, i.e., whether the awareness mechanism works with *past* or *present* activities and locations. We do not consider predicting future activities and location, since this would be forecasting, not an awareness mechanism—rather, we expect users to be able to predict what the other users are doing by studying their present and past activities (e.g. "*if John was at the cafeteria working on our paper 10 minutes ago, he might still be there and might want to talk to me about the paper*").

The list of awareness elements was then classified according to the multiuser interaction contexts shown in Fig. 3. Table 2 shows the result of this classification, indicating also when the awareness mechanism should provide present information (labeled as "Pres"), past information (labeled as "Past"), or both of them.

Table 1. Summary of awareness types to support mobile collaboration.

Awareness Type	Definition	Examples
Physical Location	Location of a user in a map.	Google Latitude [8]
Physical proximity	Whether the user is in the same physical place as another.	Hummingbird [16], Rococo [25]
Distance	Location of user in relation to other users.	Loopt [19]
Place	Location of user in a place (e.g. "cafeteria", "library").	Foursquare [6]
Movement	Direction and speed of a user with regards to other users.	Waze [31]
Profile	Shares the user profile information with other people, including the user role.	Facebook [5], Gatsby [7], LinkedIn[18]
Visibility	Indicates if the presence of the local user is visible or not to others.	Skype [27]
Availability	Indicates whether the user is busy or available to collaborate with co-workers.	Skype [27]
Activity	Indicates the activities the user is engaged in at his device.	ConNexus [29], CenceMe [20]
Connection	Indicates whether the user is connected or not.	MSN [21], Google Talk [9]
Network connectivity	The system informs when the network connectivity is lost or recovered.	Skype [27]
Message delivery	The system informs the user when her/his messages are received by the target users.	WhatsApp [32]
View	Provides visual information from a remote environment.	Skype [27], Tango [30]
Resources Accessibility	Indicates whether a resource is shared for a group, public or private.	Dropbox [3]

To perform this classification, we used as a starting point a questionnaire that asked 170 engineering students about what presence awareness mechanisms were most useful in the four different quadrants [15]. The students' strongest preferences for each quadrant are highlighted in boldface in Table 2.

Then, we interviewed over 60 people to ask them what types of awareness are most useful in each multiuser interaction situation. Based on previous experience in the design of mobile collaborative tools, as well as using the results of the questionnaires and interviews mentioned above, we developed the final classification of awareness types in the proposed four quadrants. Overall, we note the second step of the AIMMA method analyzes the multiuser interaction situation specified in the first step and suggests awareness components that could be useful to support the collaborative process (i.e. quadrants in Fig. 3).

Table 2. Recommendation of awareness components to support interaction in the four quadrants

	Reachable	Unreachable
Simultaneous	- Physical location (Pres) - Physical proximity (Pres) - Distance (Pres) - Place (Pres) - Movement (Pres) - Profile (Pres) - Visibility (Pres) - Availability (Pres) - Activity (Pres) - **Connection** (Pres) - Network connectivity (Pres) - Message delivery (Pres) - View (Pres) - Resources accessibility (Pres)	- **Physical location** (Pres-Past) - Physical proximity (Pres-Past) - Distance (Pres-Past) - Place (Pres-Past) - Movement (Pres-Past) - Profile (Pres-Past) - Visibility (Pres) - Availability (Pres-Past) - Activity (Pres-Past) - Connection (Pres-Past) - Network connectivity (Pres) - View (Pres) - Resources accessibility (Pres)
Non-simultaneous	- Profile (Pres) - Visibility (Pres) - Activity (Past) - **Connection** (Past) - Network connectivity (Pres) - Message delivery (Pres) - Resources accessibility (Pres)	- Profile (Past) - Visibility (Pres) - Activity (Past) - **Connection** (Past) - Network connectivity (Pres)

3.3 Analyzing the Collaborative Mobile Application

The *third step* involves implementing the proposed awareness mechanisms in the application under development. Naturally, in case some awareness mechanisms have already been incorporated into the application, we must first review it to determine whether the suggested awareness components are already present or not. To do that, developers will have to simulate/theatricalize the multiuser interactions that may occur during collaborative activities (i.e. those indicated in the MCM graph) and determine if in those situations the system provides any awareness mechanisms suggested by the AIMMA method.

In case a particular awareness component is not present in the application, the method suggests in which context the component should be made available. Thus, the method aims to improve contextualized collaboration support. However, the developer should ultimately decide if a certain awareness component should be included in the application or not, taking into account other factors such as, for instance, the implementation cost. In order to reduce the effort applying the proposed method, a software tool was developed. Next section briefly describes this tool.

4 A Tool for Applying AIMMA

We developed a tool supporting the AIMMA method. The tool helps software developers to create an MCM graph that describes the collaboration processes they are trying to support. The tool also allows users to generate a list of awareness requirements for the relationships defined in the MCM. For example, Fig. 4 displays a collaborative process for construction inspections we developed with the AIMMA tool. If the user chooses the "*Analyze awareness requirements*" option, the tool automatically generates the corresponding awareness requirements for each role relationship.

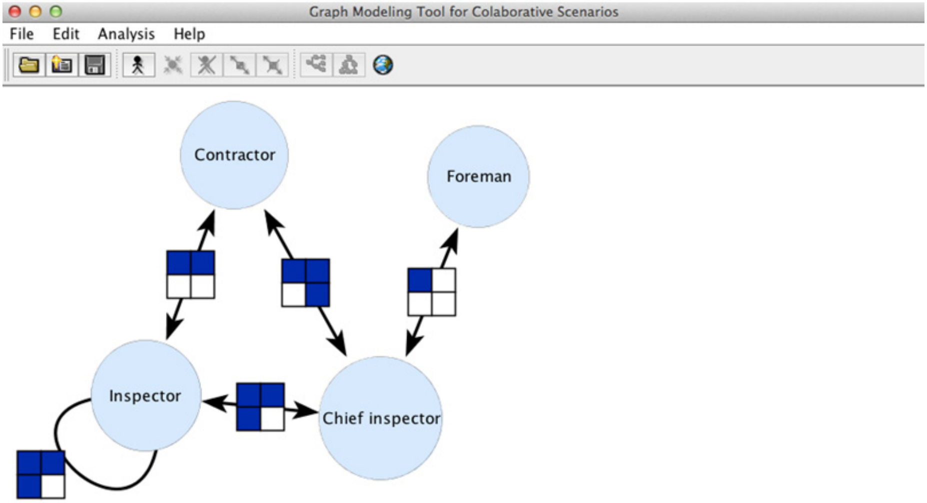

Fig. 4. Example mobile collaborative process

Fig. 5 displays the list of requirements generated for the Foreman role, which, according to the MCM graph, only interacts with the Chief Inspector. The tool shows the suggested mechanisms and allows adding other mechanisms or disregarding some suggestions by unchecking a box.

Then, the developer must review the application (or the requirements specification if the application is under development) to identify which recommended awareness mechanisms are already included in the application. The developer may indicate whether the awareness mechanism has been implemented or not, by checking the "done" box. It is important to note that each pair of multiuser interactions will generate its own list of awareness requirements. For instance, the *message delivery* requirement applies to the Foreman and Chief Inspector interactions but not to Contractors and Inspectors.

Concerning the awareness mechanisms that have been suggested by the tool but are not implemented in the application, the developer must decide if it is convenient or not to include them. To do that, the developer can use his/her own criterion and also ask the users/clients for their pertinence.

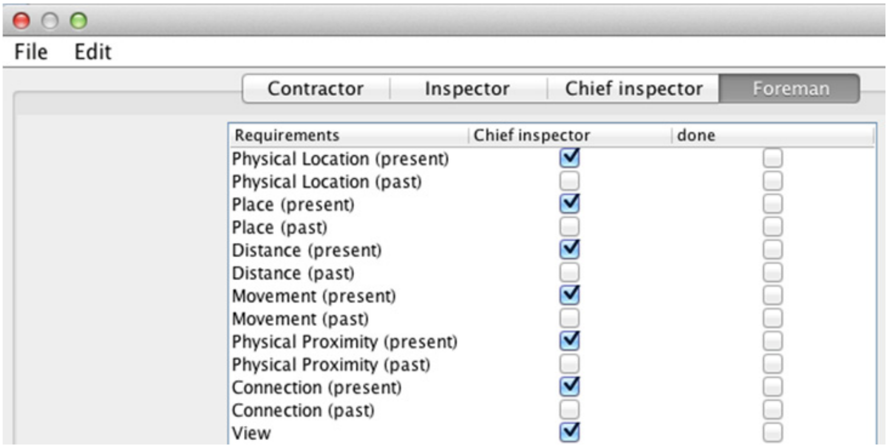

Fig. 5. Recommendation of awareness components to support users interaction

5 Case Study

The AIMMA method was used to improve the collaboration support provided by a mobile collaborative application used to perform construction inspections. The application, named COIN (Construction Inspector) [22], had a development team in charge of evolving the solution according to the users' requirements and also the opportunities identified by AIMMA. AIMMA was applied to COIN v1.5, and the obtained feedback was considered in the development of COIN v2.0. Next sections describe the initial COIN, the results obtained after applying AIMMA to COIN v1.5, the new version of COIN and some preliminary results.

5.1 The Initial COIN Tool

COIN allows a team of inspectors to record incidences in digital blueprints that represent the physical facilities of a construction project. These records are shared and discussed by several inspectors to determine whether incidences must be sent to the main contractors or subcontractors. Figure 6 shows the main user interface of COIN v1.5. The user list (also known as "buddy list") shows that two inspectors (Juan and HP-PDA) are participating in the inspection process of a building. Moreover, we can see the users' current location on the digital blueprint they are using to record incidences. This allows them to perform quick face-to-face interactions when they have to discuss an incidence record or have to coordinate their activities.

COIN also includes an instant messaging tool that allows exchanging messages among the participants in an inspection process. Connectivity among the participants is provided by a Mobile Ad hoc Network (MANET). Therefore, a wireless communication infrastructure is not required to connect the team members. A user can set his/her connection mode as "collaborative" if he/she wants to remain connected to the MANET. However, a user can set the connection mode to "stand-alone" and in that case will be disconnected from the other team members.

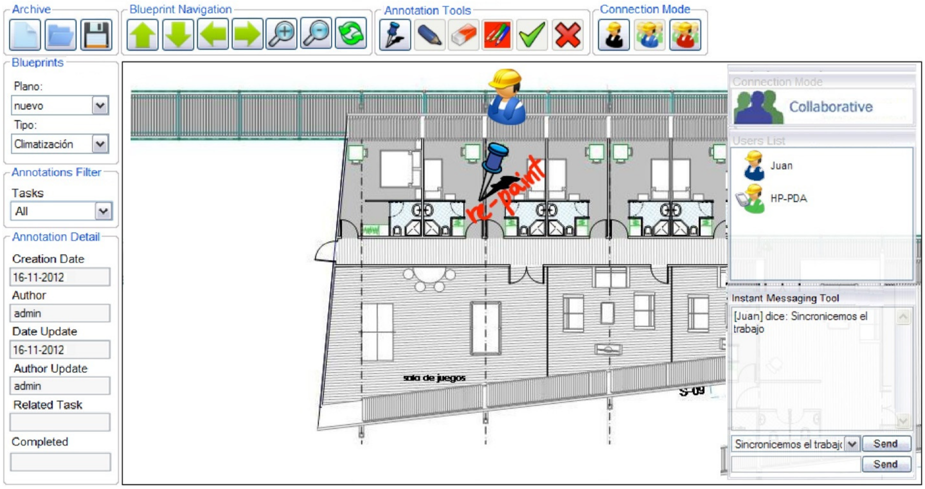

Fig. 6. Main user interface of COIN v1.5

5.2 Analyzing COIN's Awareness Support

To determine how suitable the awareness support of COIN v1.5 is, a developer created and validated the multiuser interaction graph for the inspection process (i.e. performed the first step of AIMMA). The graph is presented in Fig. 4.

Using the AIMMA tool, the developer obtained a list of awareness mechanisms that could be used to ease the multiuser interactions among specific pairs of roles. Fig. 7 shows part of the recommendations related with the Inspector role; and also shows what awareness mechanisms are already supported for that role.

AIMMA gave 23 recommendations for the Inspector's relationship with the other 3 actors. Eighteen of them were found appropriate by the developer. The awareness on place (present and past), view (present), profile (past) and resources' accessibility (present) were not considered suitable to support the construction inspection process. Four of the suggested awareness mechanisms were incorporated into COIN: physical location (present), connection (present), profile (present) and visibility (present).

Moreover, it was found that the current version of COIN does not provide awareness about the interaction between Inspectors and Contractors. After analyzing the recommendations not included in COIN, the developer decided to support the interactions among Inspectors and between Inspectors and the Chief Inspector. The resulting application is detailed in the next section.

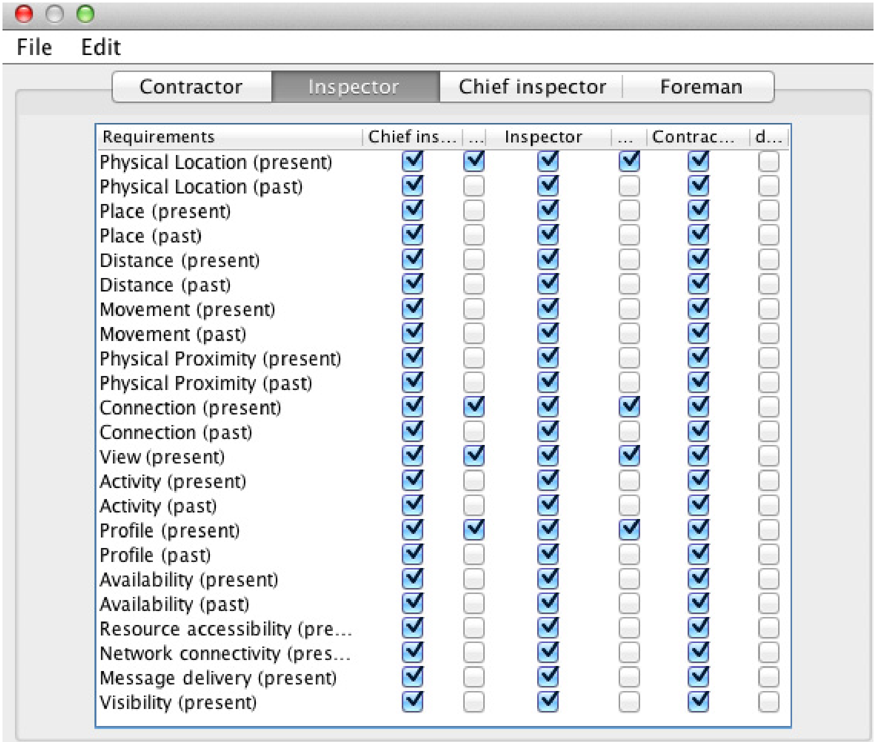

Fig. 7. Awareness support of COIN v1.5 for a user inspector

5.3 The Current COIN Tool

Figure 8 shows the user interface of COIN v2.0, which includes most of the awareness mechanisms suggested by AIMMA. Users' physical location awareness was improved by including a label with the user names, age and positions. COIN also includes a visual track that allows inferring the users' movements. The buddy list was improved to include awareness information on users' connectivity (present), activities (present and past), relative distance/proximity to other users, and also the availability (present) of other inspectors, including the Chief Inspector. The local user visibility was redefined. Now, when the user decides to be "invisible" he/she remains connected, but the presence is not visible to the other team members.

5.4 Evaluation Results

This version of COIN has been used in a real scenario to support simulated inspections, as a way to evaluate collaboration capabilities that the new features provide to the users [26]. The inspection was done in a large construction project that was at an intermediate stage. The participants in this evaluation process were a Chief Inspector and four regular Inspectors, all of them familiar with the use of COIN v1.5. Three observers were also participating in the process; one of them followed the Chief Inspector during the whole experiment.

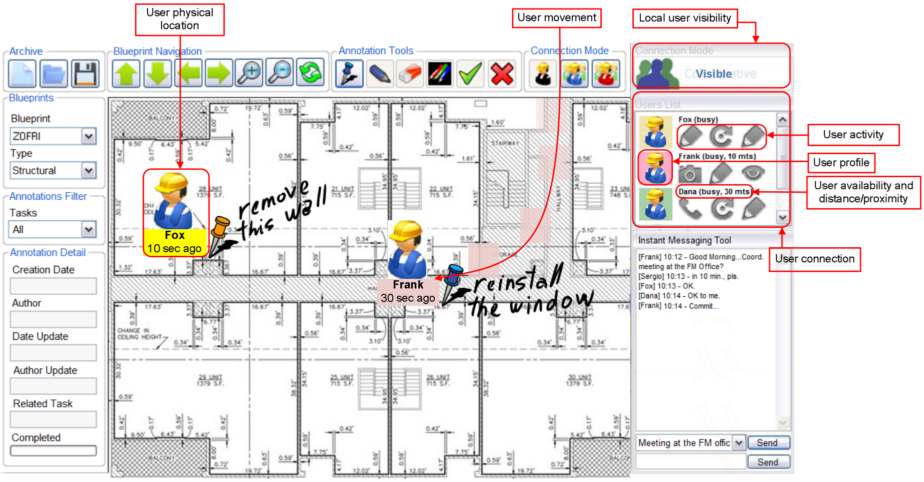

Fig. 8. Main user interface of COIN v2.0

The inspection team reviewed the electrical network of two floors of the building. Fifty post-its were placed in the physical infrastructure to simulate electrical contingency issues. Each post-it included one or more sentences describing the contingency, which were recorded by the inspectors on a digital blueprint using COIN. The Chief Inspector used COIN also to identify the position of the team members and estimate the advance level of the inspection process. The Chief used the users' activity and location awareness to find the inspectors that were delayed with the reviewing process, and thus he helped them to finish the assignment.

When inspectors needed to discuss with a teammate about a particular electrical contingency or when they had to report their inspection results, they used the movement and location awareness to find colleagues or the Chief Inspector.

After the inspection process we conducted a focus group with the participants to try to understand the impact the use of the new system had on both the people interaction process and the activity performance. The participants agreed that the awareness information embedded in the system allowed them to coordinate the tasks and get a more comprehensive view of the process. However they believe it would be even better if some context information is delivered through alarms; e.g. when an inspector finishes the assigned activity. All participants felt highly comfortable using the new version of COIN.

Although the evaluation is still preliminary, the obtained results indicate that the awareness mechanisms embedded in the new version of COIN ease interactions among Inspectors and also between the Inspectors and the Chief Inspector.

6 Conclusions and Future Work

It is well known that designing mobile collaborative applications represents a challenge for software engineers. Particularly, the design of the awareness support embedded in the system will impact the collaboration capability of the users. Moreover, the mobility of these users changes the interaction context between them, therefore the awareness support should be provided according to the interaction context that characterizes the situation that involves the potential collaborators.

This article presented the AIMMA method that helps designers of mobile collaborative applications to identify awareness mechanisms to support nomadic users that perform a particular collaborative activity. This method uses an interaction graph to determine the awareness support that is potentially useful to ease or promote collaboration between the participants, depending on their roles in that activity.

The proposed method was used to determine the awareness support of a mobile application that eases the collaborative work of construction inspection teams. The obtained feedback was then used as an input for the design of a new version of the tool. The results of the new system evaluation indicated that the awareness elements introduced in the application (according to the AIMMA suggestions) were useful and usable for the end-users.

Although these results are still preliminary, they indicate that the suggestions provided by the proposed method can be used to improve the collaboration support of mobile collaborative systems. Clearly, more experimentation is required to determine the real contribution and limitations of this proposal.

As part of the future work we will continue using the AIMMA method to help improve these applications. The other topic deserving of in-depth study is the problem of user privacy: the tradeoffs between providing awareness and granting users privacy is well known. The AIMMA method allows the designer to uncheck an awareness suggestion if the designer believes it to violate a user's privacy, however, this is a complex task and its implications and mechanisms require further study.

The AIMMA method can be adapted to help identify other services that should be required to support mobile users work. Therefore, our research on this method will continue extending it to include the identification of user interaction and data sharing services.

Acknowledgements. This work has been partially supported by the Fondecyt Project (Chile), grants: 1120207 and 11110056, and also the LACCIR project, grant: R1210LAC002.

References

[1] Antunes, P., Herskovic, V., Ochoa, S., Pino, J.A.: Structural dimensions for collaborative systems evaluation. ACM Computing Surveys 44(2) (2012)

[2] Dix, A., Finlay, J., Abowd, G., Beale, R.: Human-Computer Interaction, 2nd edn. Prentice Hall (1998)

[3] Dropbox, `http://www.dropbox.com/` (last visit: May 09, 2013)

[4] Edwards, W.K., Mynatt, E.D., Petersen, K., Spreitzer, M.J., Terry, D.B., Theimer, M.M.: Designing and Implementing Asynchronous Collaborative Applications with Bayou. In: Proceedings of Tenth ACM Symposium on User Interface Software and Technology (UIST 1997), Banff, Alberta, Canada, October 14-17 (1997)

[5] Facebook, http://www.facebook.com/ (last visit: May 08, 2013)
[6] Foursquare, http://foursquare.com/ (last visit: May 08, 2013)
[7] Gatsby, http://meetgatsby.com/ (last visit: May 08, 2013)
[8] Google Latitude, http://www.google.com/latitude/ (last visit: May 08, 2013)
[9] Google Talk, http://www.google.com/talk/ (last visit: May 08, 2013)
[10] Grudin, J.: Groupware and social dynamics: eight challenges for developers. Communications of the ACM 37(1), 92–105 (1994)
[11] Grudin, J., Poltrock, S.: Computer-Supported Cooperative Work. In: Soegaard, M., Dam, R.F. (eds.) The Encyclopedia of Human-Computer Interaction, 2nd edn., Aarhus, Denmark (2013), http://www.interaction-design.org/encyclopedia/cscw_computer_supported_cooperative_work.html (last visit: May 08, 2013)
[12] Heinrich, M., Grüneberger, F.J., Springer, T., Gaedke, M.: Reusable Awareness Widgets for Collaborative Web Applications - A Non-Invasive Approach. In: Brambilla, M., Tokuda, T., Tolksdorf, R. (eds.) ICWE 2012. LNCS, vol. 7387, pp. 1–15. Springer, Heidelberg (2012)
[13] Herskovic, V., Ochoa, S.F., Pino, J.A.: Modeling Groupware for Mobile Collaborative Work. In: Proc. of the 13th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2009), Santiago, Chile, April 22-24, pp. 384–389. IEEE Press, Los Alamitos (2009)
[14] Herskovic, V., Ochoa, S.F., Pino, J.A., Neyem, A.: The Iceberg Effect: Behind the User Interface of Mobile Collaborative Systems. Journal of Universal Computer Science 17(2), 183–202 (2011)
[15] Herskovic, V., Neyem, A., Ochoa, S.F., Pino, J.A., Antunes, P.: Understanding Presence Awareness Information Needs Among Engineering Students. In: Proceedings of the 2012 16th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD 2012), Wuhan, China, May 23-25 (2012)
[16] Holmquist, L.E., Flak, J., Wigstrom, J.: Supporting group collaboration with Inter-personal awareness devices. Journal of Personal Technologies 3, 13–21 (1999)
[17] Izadi, S., Coutinho, P., Rodden, T., Smith, G.: The FUSE Platform: Supporting Ubiquitous Collaboration Within Diverse Mobile Environments. Automated Software Engineering 9, 167–186 (2002)
[18] LinkedIn, http://www.linkedin.com/ (last visit: May 08, 2013)
[19] Loopt, http://www.loopt.com/ (last visit: May 08, 2013)
[20] Miluzzo, E., Lane, N., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S., Zheng, X., Campbell, A.: Sensing meets Mobile Social Networks: the design, implementation and evaluation of the CENCEMe application. In: Proceedings of the ACM Conf. on Embedded Networked Sensor Systems (SenSys 2008), pp. 337–342. ACM Press (2008)
[21] MSN, http://www.msn.com/ (last visit: May 08, 2013)
[22] Ochoa, S.F., Bravo, G., Pino, J.A., Rodriguez, J.F.: Coordinating Loosely-Coupled Work in Construction Inspection Activities. Group Decision and Negotiation 20(1), 39–56 (2011)
[23] Oulasvirta, A., Raento, M., Tiitta, S.: ContextContacts: Re-Designing SmartPhone's Contact Book to Support Mobile Awareness and Collaboration. In: Proceedings of MobileHCI 2005, September 19-22 (2005)
[24] Pinelle, D., Gutwin, C.: A Groupware Design Framework for Loosely Coupled Workgroups. In: Proceedings of the 9th European Conference on Computer-Supported Cooperative Work, pp. 65–82 (2005)

[25] Rococo, http://www.rococosoft.com/ (last visit: May 08, 2013)
[26] Rodríguez-Covili, J., Ochoa, S.F.: A Lightweight and Distributed Middleware to Provide Presence Awareness in Mobile Ubiquitous Systems. Science of Computer Programming 78, 2009–2025 (2013)
[27] Skype, http://www.skype.com/ (last visit: May 08, 2013)
[28] Sun, C., Xia, S., Sun, D., Chen, D., Shen, H., Cai, W.: Transparent adaptation of single-user applications for multi-user real-time collaboration. ACM Transactions on Computer-Human Interaction 13(4), 531–582 (2006)
[29] Tang, J.C., Yankelovich, N., Begole, J., Kleek, M.V., Li, F., Bhalodia, J.: ConNexus to Awarenex: Extending awareness to mobile users. In: Proceedings of the ACM Conf. on Computer-Human Interaction (CHI 2001), pp. 221–228. ACM Press (2001)
[30] Tango, http://tango.me/ (last visit: May 08, 2013)
[31] Waze, http://www.waze.com/ (last visit: May 08, 2013)
[32] Whatsapp, http://www.whatsapp.com/ (las visit: May 09, 2013)