Modeling Groupware for Mobile Collaborative Work

Valeria Herskovic, Sergio F. Ochoa, José A. Pino
Department of Computer Science, Universidad de Chile, Chile
{vherskov, sochoa, jpino}@dcc.uchile.cl

Abstract

The complexity of modeling mobile groupware systems is well known. The mobility of the collaborators, the diverse technologies available to support them and the continuous change in the collaboration scenarios bring new challenges to the design of these applications. Currently there are few languages explicitly designed to model these groupware systems. This paper presents the Mobile Collaboration Modeling language, a graphical language to represent loosely coupled mobile collaborative work. This language may be used to communicate work practices, determine user requirements and design a groupware application that accurately supports collaborative mobile work. This article presents a case study illustrating the suitability of the proposal.

Keywords: Modeling Language, Design of Mobile Groupware.

1. Introduction

Advances in mobile hardware and the implementation of wireless networking technologies such as WiMAX and 3G, are opening new possibilities to use these technologies to support various collaborative activities [4]. Mobile workers in fields such as construction, health services, education and emergency response could benefit from the communication, coordination and cooperation capabilities that mobile collaborative software may afford them.

However, mobile collaboration processes have not been studied extensively and it is difficult for software designers and developers to analyze and model this type of work. Traditional software analysis methods do not consider several important aspects of mobile collaboration such as user mobility, connection flexibility and low interdependence.

Several initiatives have provided guidelines for the design of user interfaces supporting collaboration with mobile devices [14, 18]. On the other hand, several authors have discussed the challenges of implementing back-end services to permit communication, collaboration and access to the shared environment [1, 15, 19]. Recently, some general requirements [10], contextual elements for design [2] and design patterns have been proposed to deal with these challenges [12, 13]. Unfortunately there are few guidelines to design the back-end of mobile groupware; thus, the success of this activity largely depends on the designers’ skills. Developers could benefit from an analysis technique to model the collaboration process and help them design the required application back end services.

This paper presents a modeling language that may be used during the analysis and design phases to characterize a mobile collaboration process. This language may help developers understand the collaboration scenario and therefore determine back-end and user requirements for the application that will support the collaboration process.

The next section characterizes mobile collaborative work and describes the collaborators’ interaction scenarios. Section 3 presents the related work. Section 4 describes the modeling language and its main components; it also describes how to use it in a case study. Finally, section 5 contains the conclusions and future work.

2. Mobile Collaborative Work

Collaborative mobile work is usually loosely coupled [5]. Therefore, mobile users must be able to work autonomously and carry out sporadic on-demand collaboration processes (Fig. 1) [7, 15, 16, 21]. After engaging in collaboration, users will return to autonomous work.

Figure 1. Mobile collaboration work pattern

We will characterize the periods of involvement as to clarify the possible scenarios of collaboration in which two actors may be when they need to interact. Time and space are important when designing groupware. For example, Ellis et al. proposed a classification of collaborative work based on time and
space, with synchronous/asynchronous and co-located/distributed dimensions [6]. However, in the mobile collaboration scenario, users’ locations are constantly changing, thus the line between remotesness and co-location becomes blurred. Hence, we 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., actors can expect a response in a certain period of time [9]. We define two actors are simultaneously present if they are available to work synchronously.

We can classify the possible interaction scenarios between two actors 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) (Fig. 2). Whenever the actors decide to collaborate they will be in a particular quadrant of the classification.

![Figure 2. Classification of collaboration scenarios](image)

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, work is practically autonomous and very weakly interdependent, however, providing technological support may ease the collaboration process.

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.

3. Related Work

There are several techniques that may be used to model complex systems to aid in software design. Formal models such as input/output automata may be enhanced to support the modeling of dynamic systems [3]. Agent-based models may also be used to model large-scale complex systems with several tightly coupled interacting subsystems, including human operators [11].

Constructing models for groupware requires adapting analysis techniques and methods to collaborative work. Several groupware modeling techniques focus on task analysis. Two well-known techniques are Groupware Task Analysis (GTA) [22] and Collaboration Usability Analysis (CUA) [20]. GTA aims to model the task domain for groupware, specifying agents, tasks, objects and the environment. CUA is oriented towards collaboration, using the mechanics of collaboration to specify tasks. This technique represents individual tasks and periods of teamwork, displaying loosely coupled work, although not necessarily mobile work. Petri nets and Interactive Cooperative Objects may also be used to model groupware [17]. Garrido and Gea [8] use a colored Petri net extension of task-based approaches to model cooperative systems using UML notation.

However, groupware modeling techniques alone may not be suitable to model loosely coupled mobile group work. For this type of work, models should incorporate the notions of loose coupling and flexibility in location, connectivity and involvement in the collaboration.

Pinelle [21] proposes an analysis technique designed for loose coupling, to provide an overview of important features of the collaboration scenario. This technique includes five models: the interaction model, awareness model, coordination model, group task model and loose coupling checklist. These models were developed while working on a mobile groupware application for home care workers.
4. Mobile Collaboration Modeling

This paper proposes the Mobile Collaboration Modeling (MCM) language as a graphical nomenclature that may be used to model the groupware services required to support loosely coupled mobile collaboration. It is based on a directed graph, in which the nodes represent the roles of the actors participating in the collaboration process, and the arcs represent the possible interactions among them. The analysis of an MCM graph allows groupware designers to know which functionalities should be embedded in the software application supporting the mobile collaboration process for the work scenario.

4.1 Language Specification

There are two types of nodes: those that represent roles and those representing intermediaries in the collaboration. Intermediary nodes are passive participants that do not collaborate, but they enable others to collaborate. An example intermediary node is a server with which users synchronize their work.

![Figure 3. Types of arcs](image)

A sends information to B

A and B exchange information

Arcs are used to represent interaction between two roles. A directed arc from A to B represents that role A requires sending information to role B (Fig. 3); otherwise, if no collaboration is considered, then there will be no line between them.

<table>
<thead>
<tr>
<th>Label</th>
<th>Meaning</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>Simultaneous Reachable</td>
<td></td>
</tr>
<tr>
<td>SU</td>
<td>Simultaneous Unreachable</td>
<td></td>
</tr>
<tr>
<td>NR</td>
<td>Non-simultaneous Reachable</td>
<td></td>
</tr>
<tr>
<td>NU</td>
<td>Non-simultaneous Unreachable</td>
<td></td>
</tr>
</tbody>
</table>

Each arc is labeled to represent the possible scenarios in which interaction will take place, which correspond to the previously described classification scheme. Each arc may have a simple label (SR, SU, NR, or NU), represented in Table 1, or a composite label, which is an OR composition of simple labels and represented by a square with the corresponding colored quadrants. For example, a representation showing a colored top half means a (SR OR SU) composite label.

The language models the roles and interactions of a particular collaboration scenario using this graphical nomenclature. MCM graphs may be used to distinguish each role that is present in the collaboration, communicate work practices, and determine user requirements. This may help developers to understand how collaboration takes place and design a suitable application.

4.2 Constructing an MCM Graph

In this section we describe how to generate an MCM graph that models a collaborative work scenario. A brief description of the steps for MCM graph generation is presented in Table 2.

First, the roles involved in the collaboration are identified, including any servers that are used to store and synchronize work. Second, each role is described, detailing the following information: the location where they carry out work, devices used, working hours and type of network access. Finally, the information each user must send and receive is taken into account to build the arcs and define their direction.

### Table 2. Steps for MCM Graph generation

<table>
<thead>
<tr>
<th>Step</th>
<th>Goal</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identification of roles</td>
<td>System users, servers</td>
</tr>
<tr>
<td>2</td>
<td>Role characterization</td>
<td>Location, Devices,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Working hours, network</td>
</tr>
<tr>
<td>3</td>
<td>Identification of relationships</td>
<td>Information flows</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(inputs and outputs for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>each user)</td>
</tr>
</tbody>
</table>

4.3. Identifying the Collaboration Requirements

This section describes how to use an MCM graph to aid in building an application supporting the work scenario.

In [10], we presented a list of requirements for mobile collaboration, gathered through the experiences of several researchers as well as our own. Figure 4 presents a more detailed list of these requirements organized according to the collaboration scenario in which they are present. Each quadrant (reachable/simultaneous, unreachable/simultaneous, reachable/non-simultaneous and unreachable/non-simultaneous) has a set of requirements specific to that scenario. For example, when two users are simultaneous and reachable, they should have access to online awareness information such as their status and location. Transitions between scenarios also have requirements, e.g., when users become disconnected, they should have awareness of the transition between states.

We may combine the MCM graph describing a collaboration process with the list of requirements to aid...
in the implementation of a mobile application to support collaborative work. The MCM graph displays the collaboration scenarios two users may go through, and Figure 4 displays the requirements that should be considered in those scenarios.

![Figure 4. Mobile collaboration requirements](image)

4.4. Case Study

This section presents a case study for the use of MCM graphs. The scenario is briefly described, followed by a description of the construction process of the MCM graph that describes this particular scenario and the use of the graph to help design the mobile application.

4.4.1 Scenario Description. One of the scenarios for the use of mobile collaborative technologies is the case of firefighters conducting search and rescue operations during an emergency such as a fire or building collapse. In these situations, firefighters generally use radio communication and face to face encounters to exchange information about resource allocation, location of injured or trapped people, and dangerous and safe areas.

In this scenario, teams of firefighters led by a team captain constantly move around the disaster area to assess the emergency situation. Their goal is to find and rescue all victims as fast as possible. There are several firefighters who support the search and rescue operation outside the emergency area (Fig. 5):

- **Logistics**: a group of people in charge of resource allocation.
- **Entry control**: a person in charge of recording information about each firefighter entering the disaster area.
- **Medic**: a group of people in charge of treating medical injuries and distributing victims to hospitals.
- **Incident Commander (IC)**: the person who is in charge of the whole operation. The IC must make decisions and create strategies to deal with the emergency.
- **Dispatch Center**: several people at a central facility. The IC may communicate with them to request resources or services, e.g., police support. The dispatch center is always open.

![Figure 5. Firefighters case study](image)

The firefighters’ work may benefit from the use of lightweight mobile devices they can use in addition to their typical voice communication devices. These devices may be used to mark relevant areas or points of interest, to mark the location of resources and groups of firemen, and to exchange this information. In the context of firefighters’ work during emergencies, communication infrastructure may not exist; therefore, mobile devices should form Mobile Ad-hoc Networks (MANETs). All communication conducted through the network devices should be redundant to the usual communication media.

4.4.2 MCM Graph. In this section we describe how to generate an MCM graph modeling the situation described in the previous section, following the steps shown in Table 2.

First, we must identify the roles that are involved in the collaborative scenario. The roles are: team captain (who will have face-to-face communication with firefighters on his team), logistics, incident commander, medic, entry control and dispatch center. An intermediary between the IC and the rest of the roles...
will be established in the communications area to receive all information from the emergency scenario and synchronize it with the IC.

Second, each role is characterized, describing location, devices used, working hours and network access. In this scenario, all roles (except for the dispatch center which is always active) work from the start of the emergency until its end, so work is mostly simultaneous. Location is variable throughout the emergency and support areas. Each person uses PDA devices to receive, send and synchronize information, and as it was mentioned above, network type is a MANET.

With the information from the first two steps, we create the nodes corresponding to each role. Next, we build the arcs between the roles and define their direction. For this, we study the information flows between actors. For example, a team captain must coordinate efforts between different teams, report on his team’s progress to the IC, and request resources and receive replies from the firefighters responsible for logistics. Therefore, team captains must collaborate among themselves, collaborate with logistics, and send progress reports to the communications server (which will be synchronized with the IC’s information). The resulting MCM graph is shown in Figure 6.

4.4.3 Application Design. This section describes how to use the created MCM graph to help determine the system back-end requirements and design the mobile application.

Let us study the case of collaboration between team captains. The MCM graph for the firefighters case scenario shows that while two team captains work they may be either in the simultaneous/reachable or simultaneous/unreachable scenarios. Therefore, we may infer from the requirements presented in Fig. 4, that the application may need the requirements listed in those scenarios as well as those in the transition between them.

For example, transition awareness is required because team captains will transition from a SR to a SU state. This requirement is translated into an application feature that alerts a team captain whenever another team captain is connected and available, and when a team captain has disconnected and the time and location when this happened. The possible design of this feature for a PDA display is shown in Fig. 7.

5. Conclusions and Future Work

Developing mobile applications to support loosely coupled collaborative work is a challenging task, since it is a recent area and software must overcome challenges both in design, technical implementation and adoption by users. This paper proposes a modeling language to characterize a mobile collaboration scenario to help developers clarify, analyze and design mobile groupware systems.

Pinelle [21] proposed some requirements for analysis methods for loosely coupled workgroups to be used in system design. The requirements are to capture coupling, work, communication, coordination and information utilization patterns, and the reasons for loose couplings and outcomes associated with its adoption. MCM graphs address some of the proposed requirements in the following way:

- **Coupling patterns**: MCM graphs incorporate information about relationships among all collaborators. These graphs do not include stakeholders outside of the collaboration.
- **Work patterns**: MCM graphs do not display tasks, but they do show points of shared work. The characterization of roles associated to an MCM graph presents work locations and work artifacts.
- **Communication, coordination, and information utilization patterns**: MCM graphs may be used to capture current collaboration patterns. They allow developers to find potential improvements to the collaboration as well as breakdowns in the collaboration (e.g., when two roles are in unreachable and non-simultaneous scenarios).
The next steps in this work are to implement an application to support developers when building an MCM graph. This application will also provide a quick way to check graph consistency.

Acknowledgements

This work was partially supported by Fondecyt (Chile), grants Nº 11060467 and 1080352, and LACCIR grant Nº R0308LAC004.

References


