Communication and Coordination Patterns to Support Mobile Collaboration

Roc Messeguer[1], Sergio F. Ochoa[2], José A. Pino[2], Leandro Navarro[1], Andrés Neyem[2]
[1]*Department of Computer Architecture, Universitat Politècnica de Catalunya, Spain*
{messeguer, leandro}@ac.upc.edu
[2] *Department of Computer Science, Universidad de Chile, Chile*
{sochoa, jpino, aneyem}@dcc.uchile.cl

Abstract

Wireless communication and mobile computing technologies have brought new opportunities and challenges to the CSCW area. Traditional CSCW solutions concerning communication and coordination support are not applicable to mobile collaboration scenarios. Therefore, new reusable solutions have to be proposed in order to ease the development of mobile collaborative applications. This paper presents a set of design patterns to support communication and coordination in mobile ad-hoc scenarios. These two aspects are the basis to support mobile collaboration.

Keywords: Design Patterns, Collaborative Mobile Applications, Communication/Coordination Design.

1. Introduction

A collaborative application supports the work performed by a group of collaborators who pursue a common goal. For the attainment of this goal, the groupware system must provide support in three essential aspects: communication, coordination and collaboration [5]. Communication allows interactions among the groupware applications, and the coordination aspects are in charge of governing such interactions. The collaboration process can be done by people (or software agents), who can use a groupware application to ease or promote such collaboration.

The CSCW community has designed solutions to support communication and coordination in stable communication scenarios (wired networks) and the obtained results are highly interesting. However, the current capabilities for wireless communication and mobile computing have brought new challenges for the CSCW community. Mobile collaboration is a great opportunity and a need in several scenarios [17]. A new set of solutions is required to deal with the same problems we solved for wired networks.

This new collaboration scenario does not allow the use of centralized components because the users' mobility can make these resources inaccessible. In addition, users' mobility changes the network topology and requires frequent evaluations of better paths between a consumer and a provider.

This paper proposes a set of design patterns that help to organize communication and coordination to provide mobile collaboration. Next section introduces the Mobile Ad-hoc Networks (MANETs) and the challenges to support mobile collaboration. Section 3 presents related work. Section 4 describes the patterns that are proposed to deal with some of the challenges. Finally, Section 5 has the conclusions and future work.

2. Mobile Collaboration

A mobile user is often conceived as a person executing tasks anywhere and anytime, using mobile computing devices with wireless communication capabilities. Typically they require functionalities for data synchronization and on-demand collaboration with other people. Given the uncertainty about the next collaboration scenario and its characteristics, mobile users need autonomous, flexible and interoperable collaborative solutions, regardless of the availability of centralized resources or communication infrastructure [13]. When two or more mobile users meet, their physical location should not be a limitation to collaborate. Examples of this kind of collaboration are in several areas such as: disaster relief [15], construction management [12] and health care [18].

Collaboration activities involving mobile workers are frequently supported by MANETs (Mobile Ad-hoc NETworks) [1]. However, solutions using MANETs have to be designed to support communication and coordination, by considering the features of these networks, such as small communication range, dynamic topology and high disconnection rate [4].

Some requirements influencing mobile collaboration, in terms of communication and coordination support, are the following ones:

- *Autonomy:* Collaborative mobile applications should work as autonomous solutions in terms of communication, data and functionality. This is because the MANETs have a high disconnection rate when they are used by wireless computing devices carried on by mobile users [13].

978 -1-4244-1651-6/08/$25.00 © 2008 IEEE

- *Interoperability*: Since mobile users are usually unknown persons trying to do casual or opportunistic collaboration, their collaborative mobile applications should offer communication, data and services interoperability.
- *Shared information availability:* Shared information supporting collaborative applications in these scenarios need to be highly replicated due to frequent disconnections (even using access points).
- *Variability of the work context:* Since users are on the move to carry out their activities, their work context can frequently change. Some attributes, such as MANET topology and the Internet/servers access, will change from one place to the next one.
- *Use of hardware resources:* Collaborative mobile applications should operate, in many cases, with constrained hardware resources; e.g., the case in which these solutions need to run on Personal Data Assistants (PDAs). Therefore, the communication and coordination solutions should be lightweight.
- *Low coordination cost.* Tasks are often strongly partitioned among workers. This partitioning minimizes coordination demands and it allows people to work autonomously and in parallel [16]. Ideally, the coordination process should be unattended [15].
- *Awareness of users' reachability*. Mobile workers need to know when a particular user is reachable, because they do on-demand collaboration. Hence, awareness mechanisms indicating user reachability should be embedded in mobile groupware applications.
- *Deployment ease.* An important factor is the speed of having the device ready to operate. A quick boot-up time will let workers productively use dead times. Applications – in particular, these workspaces – should also self-configure automatically after boot-up [15].

3. Related Work

There are several experiences reporting the use of collaborative mobile applications [13]. Although some of these applications are fully-distributed, they do not describe or analyze the strategies used to deal with the requirements identified in section 2. Thus, the potential design solutions cannot be evaluated when they are formalized through design patterns or reused in future application developments. Lukosch and Schümmer [10] argue that groupware reuse should focus on design reuse rather than code reuse.

A related works was done by Jørstad et al., who studied the feasibility of developing generic coordination services in a distributed (but stable) work scenario [9]. They concluded generic coordination services definition is feasible and they developed a model that was mapped to a Service Oriented Architecture. The generic coordination services they defined included Locking, Presentation Control, User Presence Management, Organization Management and Communication Control. These services were put together into a coordination service layer and made available to the collaborative applications. However, this proposal does not consider changes in the local work context and people's mobility.

There is a large list of reusable designs applicable to the coordination services design of groupware applications for fixed networks [2, 7]. However, the contextual variables influencing the collaboration scenario (e.g. communication instability and low feasibility to use servers) and the mobile work (e.g. use of context-aware services and support for ad-hoc coordination processes) make such solutions unsuitable. Next section presents the pattern we propose to structure the communication and coordination services in order to support mobile collaboration.

4. Patterns

Following the ideas of Ellis et al. [5] we propose a layered and fully-distributed architecture. It is composed of three layers and it is the base supporting the rest of the patterns (Figure 1).

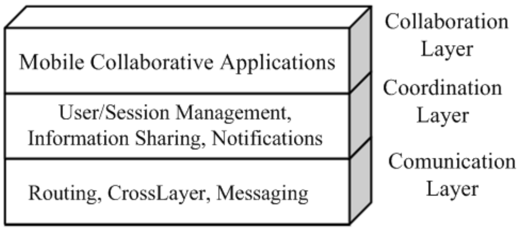

Mobile Collaborative Applications
User/Session Management, Information Sharing, Notifications
Routing, CrossLayer, Messaging
Collaboration Layer
Coordination Layer
Comunication Layer

Figure 1. Layered architecture to support mobile collaboration

The layered architecture allows mobile groupware designers to separate the software concerns. It also allows to increase the system scalability and flexibility, and gives a sound platform to deal with interoperability issues. Since this architecture is fully replicated, each mobile unit (user + groupware application) becomes autonomous in terms of services and data. Thus, we can see the collaboration scenario as a dynamic mesh where no centralized components are allowed. Next section briefly describes the proposed design patterns using the description language proposed and used in [10].

4.1. Distributed Session Management

Intent. It allows users to setup a network environment including transport layer (TCP/IP) and session layer for collaborative interactions using a MANET.

Context. Groups of users need to interact among them synchronously/asynchronously using a Mesh network or MANET. Regardless of how dynamic is the network topology, if these interactions are feasible, they should be easy to carry out and involve a low coordination cost.

Related Requirements. Low Coordination Cost, Autonomy and Ease of Deployment.

Problem. Users and groups of users using a mobile ad-hoc network need to be univocally identified in order to coordinate the interactions among them and provide an accurate message deliver/routing. However, Mesh networks and mainly MANETs do not allow centralized components able to link the real and the virtual identity. This situation constrains the groupware applications capabilities to support interactions between two users in these mobile, dynamic and changing networks.

Solution. The session management should be fully-distributed. It is not a problem because the proposed architecture is fully distributed and each mobile unit has to be autonomous. The user's virtual identity (VI) is a unique ID, which is linked to the real identity (RI) without concerning the user's device type or IP address. User sessions can be implemented as dynamic arrays of virtual identities (Figure 2).

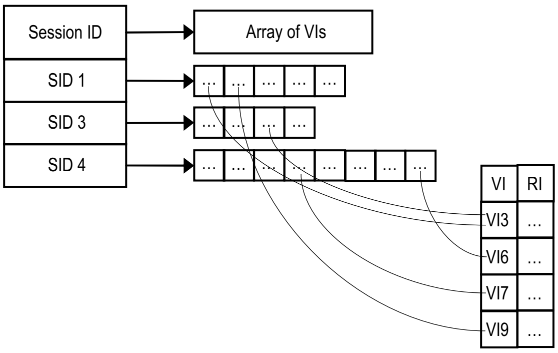


Figure 2. Basic structure of MANET session management

Each session has a unique Session ID (SID). Session members keep a local instance of the arrays involving them. These arrays need to be synchronized automatically or on-demand in order to keep the session state updated.

Each node in the MANET has a virtual ID and a set of sessions which the current user belongs to (Figure 3). This information is used to manage the session and also to route and deliver messages. This structure allows the management of multiple sessions at the same time, even if some session members get temporarily disconnected.

Related Patterns. Session [6] is not focused on how to organize a session. That pattern deals with collaborative sessions; the session instance does not include information about the applications used in the session. Collaborative Session [11] models the context for collaboration as a shared session object, visualizes the session state and supports users in starting, joining, leaving, and terminating the session and automates the selection and start of tools.

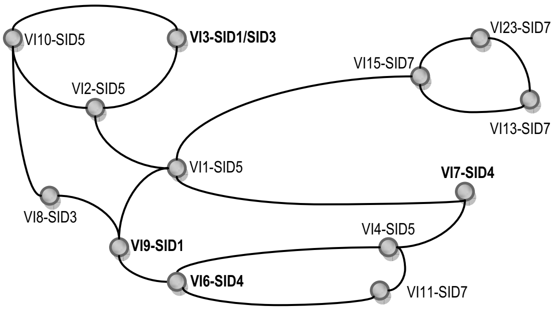


Figure 3. Information support for session management in a MANET

4.2. Work Session

Intent. This pattern allows users belonging to the same session to collaborate using a MANET.

Context. Mobile users in the same session collaborate interchanging messages and data. The setup and interaction process should be as simple and automatic as possible.

Related Requirements. Information Sharing and Low Coordination Cost.

Problem. Mobile users do not have time to spend settling up issues to collaborate. Thus, the structure of a work session should include components able to ease the interaction process. The functionality should include message delivery, information sharing and users' connection / disconnection to a session.

Solution. Every work session should be modeled as an entity composed of a session shared data space and a set of members (Figure 4). The session shared data space is composed of the public data spaces of each mobile unit currently active in the session. The session members (identified by a virtual ID) represent the set of mobile units able to interact to each other as part of a collaboration process.

Each session member has four components: (1) a private data space, (2) a public data space, (3) a list of reachable mobile units, (4) an input/output communication channel. Typically, a mobile user puts into the public data space those resources he/she wants to share with other session members. These resources can be shared through replication or synchronization processes. Private resources are kept into the private data space, which is accessible just for the local user.

Session members also locally record information about which mobile units are reachable by them. Since the network topology is continually changing based on the users' displacement, this information is very useful to deliver/route messages and to determine when a message should be transmitted because it has a high probability to get to the destination unit.

Finally, the communication process among session members is carried out through an input/output channel, which is shared by the session members. When a user gets in/out a session a predefined message is transmitted through the channel (multicast) to the list of session members. This announcement allows the user to record the mobile unit in the list of reachable mobile units that the peer keeps. Moreover, the new user receives a response of his/her peers. Such information is used to update the local list of reachable mobile units, and also to implement awareness mechanisms about users connection. Session members could also record information of the interim nodes of the MANET as a way to ease the message routing process.

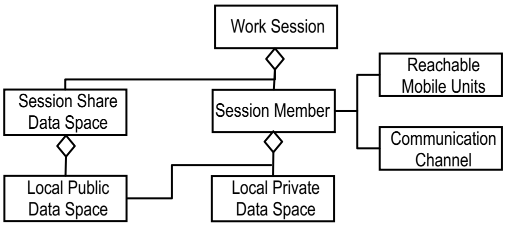


Figure 4. Structure of a work session

Related Patterns. Room [11] focuses only on users who are active in collaboration; it does not consider the com-munication capabilities and low setup cost. The new pattern deals with all user's basic needs: data sharing, user availability awareness, and message delivery.

4.3. Mobile User List

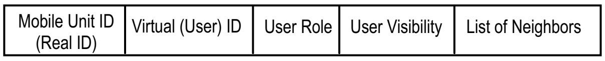
Intent. This pattern shows who is currently participating in a work session.

Context. In a work session the user presence depends on the user reachability and visibility. User presence is a well-known mechanism to motivate collaboration processes; therefore it should be provided as part of the mobile collaborative applications.

Related Requirements. Awareness of Users' Reachability, and Variability of the Work Context.

Problem. A well-known mechanism to implement user presence is to keep a list of current users. However, this list is highly dynamic in this scenario and it is distributed. Just a part of it is valid for each session member. In addition, some users could want to be invisible to other users (because of privacy purposes), but available for message routing purposes.

Solution. The list of reachable mobile units presented in the previous pattern is a data grid. Each record includes the mobile unit ID, the virtual (user) ID, the user's role, the visibility attribute and the list of neighbors (Figure 5).



Mobile Unit ID (Real ID)	Virtual (User) ID	User Role	User Visibility	List of Neighbors

Figure 5. Structure of list of reachable mobile units

Particularly, the user visibility attribute allows to implement awareness mechanisms of user availability and user roles. The list of neighbors is the set of nodes that eventually can be used as interim points to reach a (remote) mobile user. Sometimes the distance between two session members is more than one hop (Figure 2); thus, neighbor nodes should be used to support the communication process.

The list of reachable mobile units is updated by two mechanisms: (1) peers discovery and (2) list synchronization. Peers discovery involves sending a message to a peer destination using an ad-hoc gossip multicast [8]. If the destination is reached, it returns a message to the sender indicating the list of interim visited nodes. Such data is used to update the local list of reachable mobile units and neighbors. Then a change-propagation mechanism can be triggered to the rest of session members. In that case, the update to the list is done using a typical synchronization process.

Related Patterns. User List [11] only shows collaborative-active users and Session [6] only shows all the group users (connected and disconnected).

4.4. CrossLayer

Intent. This pattern allows sharing information among layers of a groupware system architecture. This data could be: a mobile unit list, the MANET topology, and features of the mobile computing devices.

Context. A groupware architecture separates concerns in different layers. Each layer provides services and records data related to such services. Interoperability among services is vital to support collaboration.

Related Requirements. Interoperability and Awareness of users' reachability.

Problem. Interoperability among services should be possible to support collaboration. No matter if the services are local or remote. This interoperability should also be provided independently if the services are provided by a Laptop, a PDA or a Smart-Phone.

Solution. Groupware services and public data structures belonging to each layer of the architecture should be accessible through an API. Important data could be shared using a mechanism to ensure services interoperability, e.g., mobile devices with few hardware resources could require lightweight mechanisms for data sharing or peers discovery. However, if an application running on a laptop must interoperate with a service running on a PDA, contextual information about the PDA's hardware resources will be needed in order to dynamically adapt the interaction protocol.

Related Patterns. The most related pattern is Layer [3] but it is not focused on distributed systems or services interoperability.

4.5. MANET Routing

Intent. This pattern allows to route messages between mobile units located at more than one hop of distance.

Context. Ad-hoc networks are not supported by infrastructure. The nodes in the network use wireless communication and a dynamic topology for data dissemination and gathering. This lets ad-hoc networks be useful in many scenarios, providing communication autonomy and interoperability to mobile workers.

Related Requirements. Awareness of Users' Reachability and Easy to Deploy.

Problem. The current wireless communication norms supporting mobility have a limited communication threshold. For example, the Wi-Fi threshold is about 200 meters in open areas and 20 meters in built areas. Most groupware solutions need to extend the communication threshold as much as possible to increase the interaction scope; hence, routing mechanisms are required to allow it.

Solution. The message delivery service should be based on ad-hoc gossip multicast [8], because it offers an intermediate solution between the routing and flooding techniques. By using gossip-based multicast, the delivery mechanism achieves high reliability with moderate degradation of performance. This delivery strategy relies on the following three-phases algorithm:

Potential Disconnection Detection: This phase is performed permanently by every node in the MANET. At every time step *t*, each node monitors its 1-hop neighbors. For each 1-hop neighbor located at a distance greater than a certain number of hops, given as a parameter, the monitoring node records such distance at that moment. At the next time step, the monitoring node will determine the new distance for each previously monitored 1-hop neighbor. If, for each 1-hop neighbor, the distance has increased, then the monitoring node will try to find out if at least one of its other 1-hop neighbors is near to the one under observation. If the monitoring node fails on such task, it will propagate a "potential disconnection" message through the MANET.

Correction: The second phase is also implemented by each mobile unit in the MANET. Once a node detects a potential disconnection, it propagates a message through the network, using gossip-based multicast. Any mobile unit receiving that message will wait a given period of time for the counterpart message; i.e., the message sent by the other node involved in the potential disconnection. If the counterpart message is received by the mobile unit, then it will ignore the situation because it means there is at least one alternative route connecting both nodes. On the contrary, if the counterpart message is not received by the unit, then it assumes a disconnection is in progress and it will set itself towards that task; i.e., move towards the potential disconnection area.

Maintenance: Each mobile unit in the potential disconnection area tries to detect the presence of the requesting nodes during the third phase of the algorithm. If a unit does not find the requesting nodes after a given period of time, then it sets itself back to Idle state. Nonetheless, if a unit detects the requesting nodes, then it sets its state to *supporting mode*. The unit will remain in such a mode until either the supported nodes are again within communication range or the supported nodes are apart and the disconnection is imminent. In such case, the mobile unit will play the potential disconnection phase.

This three-phase algorithm relies on two assumptions. On the one hand, the set of mobile units is comprised of homogenous devices not only in terms of communication capability, but also in terms of mobility; i.e., they are able to move at similar speeds. On the other hand, it is assumed the environment is free of obstacles; i.e., the time for a mobile unit to reach a target location depends only on the distance.

Related Patterns. No well-known patterns for routing in MANETs where found.

4.6. Summary

Considering the proposed patterns and the requirements for mobile collaboration proposed in section 2 it is possible to draw a correspondence matrix (Figure 6). This matrix allows to developers to select one or more design pattern in order to deal with a particular problem/limitation.

	Fully-distributed architecture	Layered architecture	Distributed session management	Work session	User list	Cross layer	Routing
Autonomy	●		●				
Interoperability		●				●	
Information sharing				●			
Context-aware					●	●	
Lightweight							
Low coordination cost			●	●			
Peers detection					●		●
Easy to deploy	●		●				●

Figure 6. Correspondence matrix

These proposed patterns have been implemented in a middleware platform [12] and several applications are currently using these reusable designs [12, 13]. Among these applications are mobile collaborative applications to support disaster relief operations, to conduct inspections in construction sites and to manage exams in computer science courses.

5. Conclusions

This paper presented a set of design patterns to deal with several challenges occurring in mobile collaboration. Although this is a partial list of patterns, it is useful enough as to implement most of the communication and coordination mechanisms required for mobile collaboration. The proposed fully distributed architecture provides autonomy to the groupware solutions and the separation of design concerns in several layers provides flexibility and scalability.

At the moment, these patterns have shown to be useful to design both, mobile groupware applications and a middleware to support collaborative systems. However, the authors have been involved in each one of these testing experiences. Thus, as future work it is required to carry out evaluations with external groupware developers. Furthermore, the list of proposed patterns should be extended and they have to be grouped in a pattern system.

Acknowledgement

This work was partially supported by Fondecyt (Chile), grant Nº 11060467 and 1080352, the Spanish MEC project P2PGrid TIN2007-68050-C03-01. The work of Andrés Neyem was partially supported by the Scholarship for Thesis Completion from CONICYT (Chile).

References

[1] R. Aldunate, S.F. Ochoa., F. Pena-Mora and M. Nussbaum, "Robust Mobile Ad-hoc Space for Collaboration to Support Disaster Relief Efforts Involving Critical Physical Infrastructure", *ASCE Journal of Computing in Civil Engineering*, 2006, 20(1), 13-27.

[2] M. Arvola, "Interaction Design Patterns for Computers in Sociable Use", *International Journal of Computer Applications in Technology*, 2006, 25(2/3), 28-139.

[3] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord and J. Stafford, *Documenting Software Architectures*, Addison-Wesley, 2003.

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

[5] C.A. Ellis, S.J. Gibbs and G.L. Rei, "Groupware: Some Issues and Experiences", Communications of the ACM, 1991, 34(1), 38-58.

[6] L.A. Guerrero and D.A. Fuller, "Design Patterns for Collaborative Systems", Proc. of CRIWG'99, IEEE CS Press, Cancún, Mexico, 1999, pp. 270-277.

[7] L.A. Guerrero and D.A. Fuller, "A pattern system for the development of collaborative applications", *Journal of Information and Software Technology*, 2001, 43(7), 457-467.

[8] Z. Haas, J. Halpern, and L. Li, "Gossip-Based Ad Hoc Routing", Proc. of IEEE Infocom'02, New York, USA, 2002, pp. 1707–1716.

[9] I. Jørstad, S. Dustdar and D. Van Thanh, "Service Oriented Architecture Framework for collaborative services", Proc. of WETICE'05, Sweden 2005, pp. 121-125.

[10] S. Lukosch and T. Schümmer, "Groupware development support with technology patterns", *International Journal Human-Computer Studies*, 2006, 64(7), 599-610.

[11] S. Lukosch and T. Schümmer, "Patterns for Session Management in Groupware Systems", Proc. of EuroPLoP'05, Irsee, Germany 2006, pp. 333-378.

[12] A. Neyem, S.F. Ochoa and J.A. Pino, "Supporting Mobile Collaboration with Service-Oriented Mobile Units", *Groupware: Design, Implementation, and Use*, 2006, LNCS 4154, 228-245.

[13] A. Neyem, S.F. Ochoa and J.A. Pino, "Designing Mobile Shared Workspaces for Loosely Coupled Workgroups", *Groupware: Design, Implementation, and Use*, 2007, LNCS 4715, 173-190.

[14] A. Neyem, S.F. Ochoa and J.A. Pino, "Integrating Service-Oriented Mobile Units to Support Collaboration in Ad-hoc Scenarios", *Journal of Universal Computer Science*, 2008, In Press.

[15] A. Neyem, S.F. Ochoa, J.A. Pino and M. Borges, "Supporting Group Decision Making and Coordination in Urban Disasters Relief Efforts", *International Journal of Decision Systems*, 2007, 16(2), 143-172.

[16] D. Pinelle and C. Gutwin, "Loose coupling and healthcare organizations: adoption issues for groupware deployments", *Computer Supported Cooperative Work*, 15(5-6), 2006, 537-572.

[17] H. Schaffers, T. Brodt, M. Pallot and E. Prinz, *The Future Workplace - Perspectives on Mobile and Collab. Working*, Telematica Instituut, The Netherlands, 2006.

[18] M. Tentori and J. Favela, "Activity-Aware Computing in Mobile Collaborative Working Environments", *Groupware: Design, Implementation, and Use*, 2007, LNCS 4715, 337-353.