Supporting Mobile Collaboration with Service-Oriented Mobile Units

Andrés Neyem, Sergio F. Ochoa, and José A. Pino

Department of Computer Science, Universidad de Chile.
Blanco Encalada 2120, Santiago, Chile
{aneyem, sochoa, jpino}@dcc.uchile.cl

Abstract. Advances in wireless communication and mobile computing extend collaboration scenarios. A current strategy to address productive, educational and social problems is to incorporate mobile workers using computing devices into work practices. Typically, collaborative applications intended to support mobile workers involve some type of centralized data or services. This situation constrains the collaboration capabilities, particularly in ad-hoc communication scenarios. We propose an autonomous software module able to provide and consume services from others units. We call it a Service-Oriented Mobile Unit (SOMU). A SOMU has been implemented as a middleware running on laptops and PDAs. Collaborative mobile applications developed on this middleware are then able to interact among them almost in any communication scenario. Availability of this tool is particularly relevant to support mobile collaboration when there is no stable communication support or no communication at all.

Keywords: Middleware for Mobile Groupware, Service-Oriented Mobile Units, Web services Platform, Ad-hoc Collaboration Scenarios.

1 Introduction

Fast development in the area of information and communication technology and especially in broadband internet access and mobile computing has changed the established ways of communication, learning, entertainment and work in professional and private lives. The *mobile* and *mobility* concepts have a strong link to wireless technologies [1]. Most often a mobile worker is conceived as a person moving and executing tasks anywhere and anytime, using mobile computing devices with wireless communication capabilities. Provided the current mobile computing devices have wireless communication capabilities, any place becomes a potential scenario to support mobile work. Examples of these scenarios are: parks, coffee shops, hospitals, universities, schools, shopping malls, offices and airports.

Mobile workers are on the move to carry out their activities. Usually, they have some instances for data synchronization or collaboration with other people. Mobile workers are frequently not sure which is the next collaboration scenario and its characteristics. Therefore, they need autonomous and flexible collaborative solutions independently of the availability of centralized resources or fixed wireless communication infrastructure (access points). When two or more mobile workers meet, the physical scenario must not be a limitation to collaborate.

Y.A. Dimitriadis et al. (Eds.): CRIWG 2006, LNCS 4154, pp. 228 – 245, 2006.

Collaboration activities involving mobile workers can be supported by mobile networks, also called MANETs (Mobile Ad-hoc NETworks) [16]. However, it means solutions including MANETs to support the computer-based collaborative activities should be designed and implemented. Most collaborative applications intended to support mobile workers involve some type of centralized data or services. This situation constrains the collaboration possibilities, particularly in ad-hoc communication scenarios. A software piece which is able to provide and consume services from others units is proposed. It is called *Service-Oriented Mobile Unit* (SOMU). The solution is fully distributed. Each unit has been implemented as a middleware running on laptops and Personal Data Assistants (PDAs). Collaborative mobile applications developed on this middleware are then able to interact among them almost in any communication scenario. Thus, mobile workers using such applications can collaborate when there is no stable communication support or no communication at all. Two application scenarios are briefly described below to illustrate the role of MANETs in mobile collaboration.

- *Disaster Relief*: Activities to resist and recover from natural, hazardous and intentional eXtreme Events (XE) are highly dynamic and demand effective collaboration among a broad range of organizations. First responders (police, firefighters and medical personnel) deployed in the work area need to know the information about the site and affected buildings (e.g. maps, probable people locations and vulnerable points), exit routes, resources deployed in the area and tasks assignment. Mobile workers from several organizations need to be autonomous, interoperable and carry diverse shared information to do the assigned activities. Sometimes they also need to update such information and communicate the updates to the partners, leaders and other organizations in order to support decision-making processes. Typically, this collaboration scenario has minimal or no communication capabilities [6]. However, collaboration among first responders is required. Government authorities in charge of macro-decisions should be able to access information from the mobile workers (e.g., police, firefighters and medical personnel) to monitor the activities evolution and make corrections on previously made decisions.
- *Building and Construction*: The building and construction industry is characterized by: (a) dispersed teams working on the development of a new site, (b) teams do not belong to the same company, (c) they are not able to use fixed communication infrastructure and (d) they need to be on the move to carry out the assigned work. For example, electrical engineers (mobile workers) belonging to a company need to be on the move in order to inspect and record the status of the electrical facilities being developed by the company employees at a construction site. During the inspection, each engineer updates the information recording the current status of the electrical facilities. After the inspection and before leaving the construction site, the engineers meet to check agreement on the updated information and review it. If they detect incomplete or contradictory data, some of them can inspect the facilities again in order to solve such case. Similarly to the previous scenario, mobile workers need autonomy, interoperability and they also need to be able to collaborate no matter the features of the physical scenario.

Mobile computing devices and mobile ad-hoc wireless networks (MANETs) offer a wide range of new collaboration possibilities for mobile workers. However, the design and implementation of the mobile collaborative solutions for ad-hoc scenarios imply several challenges in terms of the following aspects.

Autonomy: Collaborative mobile applications should function as autonomous solutions. Communication availability in the physical scenario and access to centralized shared data and services cannot be a limitation to support collaboration among mobile workers in ad-hoc scenarios. Therefore, solutions able to work in Peer-to-Peer (P2P) settings are required.

Interoperability: Provided mobile workers may need to do casual or opportunistic collaboration, the collaborative mobile applications they use should offer data and services interoperability.

Shared information management: Shared information supporting collaborative applications in these scenarios need to be highly replicated since there are frequent disconnections in wireless networks (even using access points). Keeping the shared information coherence in a P2P network is not a trivial problem to solve.

Use of hardware resources: The collaborative mobile applications should operate, in many cases, with constrained hardware resources; e.g., the case in which these solutions need to run on PDAs.

Next section describes the challenges and opportunities offered by service-oriented computing to support collaboration in ad-hoc wireless settings. Section 3 presents related work. Section 4 describes the way to overcome the stated challenges with SOMUs. Section 5 shows two application scenarios, and section 6 presents the conclusions and future work.

2 Service-Oriented Computing in Ad-Hoc Wireless Settings

Ad-hoc networking refers to a network with no fixed infrastructure [24]. When the nodes are assumed to be capable of moving, either on their own or carried by their users, these networks are referred as MANETs. The nodes of the network rely on wireless communication to collaborate with each other. The advantage of ad-hoc networking is that the absence of a fixed infrastructure reduces the cost, complexity and time required to deploy the network. It also allows users to be on the move transporting their communication capabilities [23]. Unfortunately, most of these MANETs have a small communication threshold in terms of allowed distance between two mobile workers. In addition, the lack of a fixed infrastructure introduces challenges for using and maintaining ad-hoc networks. Knowledge of various factors will help to motivate understanding of the protocols that have been developed for ad-hoc networks. A brief explanation of these properties follows.

- *No pre-existing infrastructure:* By definition, ad-hoc networks do not have any infrastructure. The nodes in the network rely on wireless communication for information dissemination and gathering. This lets ad-hoc networks be used in remote environments, and mainly to support mobile workers. Moreover, the MANETs are attractive because of the reduced effort to set up and use them.

- *Small communication threshold:* Mobile computing devices provide communication services without using a base station when they are part of a MANET. Thus, each device may function as a base station to act as a gateway between peer devices or to access other networks. The current wireless communication norms supporting mobility have a limited communication threshold (or communication range). For example, the IEEE 802.11b/g (Wi-Fi) threshold is about 200 meters in open areas and 20 meters in built areas.
- *Power-scarce devices*: Mobile devices making up the ad-hoc network have a physical environment that is assumed to be devoid of resources such as power. In fact, because of the absence of any underlying infrastructure, power outlets generally are not available. For this reason, mobile devices that form the ad-hoc network use either battery power or passive power sources, such as solar energy. This fact further reduces the communication threshold of this type of networks.
- *No centralized mechanisms*: Since ad-hoc network do not have any underlying infrastructure and wireless communication is employed, centralized routing algorithms are not applicable. The cost of transmitting data from all nodes in the network to a central location becomes prohibitively expensive in terms of power usage. Furthermore, centralized components become critical failure points and then there are the typical problems with scalability and fault tolerance for processing all the information.

On the other hand, *Service-Oriented Computing* (SOC) is a new paradigm gaining popularity in distributed computing environments due to its emphasis on highly specialized, modular and platform agnostic code facilitating interoperability of systems [22]. A key issue with SOC in ad-hoc networks is to mitigate the problem of frequent disconnection and to ensure that some channel between the user and the provider of a service is maintained for a significant period. Furthermore, SOC helps decouple concerns about network availability and connectivity and it also implies simplifications in the software development process.

The service model is composed of three components: services, clients and a discovery technology. Services provide useful functionality to clients. Clients use services to support complex functionalities that will be available for users. The discovery process enables services to publish their capabilities and clients to find and use needed services. As a result of a successful lookup, a client may receive a piece of code that actually implements the service or facilitates the communication to the server offering the service. The implementations of service-oriented models may have some limitations in terms of functionality because of the peculiarities of the ad-hoc wireless settings.

The idea of using mobile computing devices as hosts for service registries is very appealing. However, overloading simple devices belonging to a work session may lead to a defensive behavior from a collaborative system, e.g., terminating advertisement broadcasts or completely ignoring client communication.

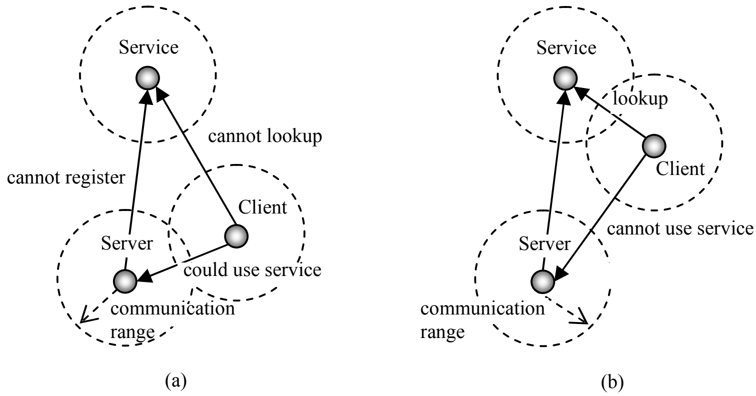


Fig. 1. a) The client could use the service but it cannot discover it because the service registry is not accessible; b) A client discovers a service which is no longer reachable

Failure of a mobile computing device implies a complete lack of communication between users in a collaborative session and between clients and services whose communication is routed via this device, even if they could communicate directly. Therefore, the service model needs to adapt itself to the new networking conditions. For example, if the node hosting a service registry suddenly becomes unavailable, the advertising and lookup of services becomes paralyzed even if the pair of nodes representing a service and a potential client remains connected (Fig. 1a). Furthermore, due to frequent disconnections and mobility of nodes, there is another problem when the advertisement of a service is still available in the lookup table until its lease expires (Fig. 1b).

As a summary, high degree of freedom and a fully decentralized architecture can be obtained in MANETs at the expense of facing significant new challenges. MANETs are opportunistically formed structures that change in response to the movement of physically mobile hosts running potentially mobile code. New wireless technologies allow devices to freely join and leave work sessions and networks, and exchange data and services at will, without the need of any infrastructure setup and system administration. Frequent disconnections inherent in ad-hoc networks lead to inconsistency of data in centralized service directories. Architectures based on centralized lookup directories are no longer suitable. Therefore, the model and technologies addressing these issues should consider all nodes as mobile units able to provide and consume services from other mobile units.

3 Related Work

Several collaborative solutions have been proposed to support mobile workers [2], [8], [17], [18], [26]. Although the proposals have shown to be useful to support specific collaborative activities, they were not designed as general solutions. Therefore, the capability to reuse these solutions in various work scenarios is relatively small.

On the other hand, there are several interesting initiatives in the middleware area, which propose reusable functions to support collaboration in P2P networks. One of them is LaCOLLA [14]. This middleware has a P2P architecture and provides general purpose functionalities for building collaborative applications. LaCOLLA works well in networks with important signal stability, such as fixed or one-hop wireless networks. However, the middleware does not support autonomous members of a group and does not have components and mechanisms that will allow mobile devices become LaCOLLA peers.

Unlike LaCOLLA, the iClouds framework offers spontaneous mobile user interaction and file exchange in mobile ad-hoc networks [11]. This framework also provides independence of a server doing a full replication of any shared file, which is appropriate in MANET scenarios. However, it does not provide support to exchange shared objects, just files. In addition, iClouds does not distinguish among copies of the same shared file (e.g. master and slave copies) and it does not support distributed operations on those files either. The functions provided by iClouds are focused just on data sharing.

There are frameworks that provide, through an API, specific functionalities to support mobile collaboration, such as YCab [5] and YCab.NET [21]. These frameworks implement their own protocol and they provide just the following generic services: session manager, text chat, image viewer, GPS and client info. Probably, the most popular framework to support P2P collaboration is JXTA [13]. This framework provides a common platform to help developers build distributed P2P services and applications. Here, every device and software component is a peer and can easily cooperate with other peers. Although JXTA has shown to be useful to support collaboration in P2P networks, it also requires a fixed or one-hop wireless network (similar to LaCOLLA). Therefore, it is not well suited to apply it in ad-hoc mobile work settings.

On the other hand, Nokia has developed a services-oriented framework that could be used to support mobile collaboration. This framework includes a set of APIs and an SDK (Software Development Kit) allowing developers to create service-oriented applications that act as consumers of Web services on mobile devices [12]. Provided the mobile applications can just consume services, their autonomy is limited and they require a service provider, which is not suitable for ad-hoc mobile work scenarios.

Currently, there are several proposals to share information in P2P networks, even considering mobile computing devices [10], [20]. Typical examples are tuple-based distributed systems derived from LINDA [7], such as: FT-LINDA, JINI, PLinda, T-spaces, Lime, JavaSpaces and GRACE [9], [19], [3]. Despite the fact these implementations work in P2P networks, they use centralized components that provide the binding among components of the distributed system. Other middleware, such as XMIDDLE [15] and PASIR [20], allow mobile hosts to share documents across heterogeneous mobile hosts, permitting on-line and off-line access to data. Nevertheless, these middleware are just focused on data sharing and they do not support the autonomy and interoperability capabilities required by mobile workers.

4 The Services-Oriented Mobile Unit

The need to support mobile collaboration in ad-hoc work scenarios and the limitations of current solutions to support it motivated the development of the SOMU software module. SOMU is a lightweight platform able to run on PDAs and notebooks. It enables each mobile computing device to produce and consume Web services from other peers. Such functionality is implemented in a lightweight Web server called *μWebServer* (Fig. 2). Thus, the autonomy and part of the interoperability required by mobile workers is supported.

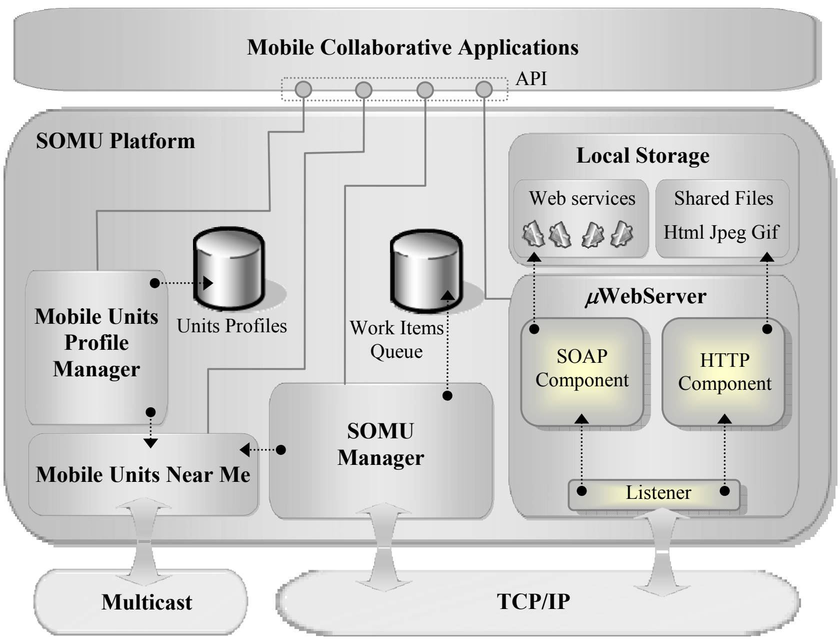


Fig. 2. SOMU Architecture

SOMU also implements a *local storage* which is composed of (1) a shared storage space to allocate the files the mobile unit wants to share, and (2) a space to allocate those Web services exposed by the mobile unit. By default, SOMU provides basic Web services for Web services description and discovery.

The *SOMU Manager* is the component in charge of creating, storing and dispatching work items when a mobile collaborative application invokes Web services exposed by other mobile units. The work items stored in a mobile unit represents the Web Services (WS) invocations that such unit needs to perform. Each work item is composed of a ticket, a mobile universal unit, the WS proxy, WS input and WS output. The *ticket* is the work item identifier. It is used to communicate the results of a WS invocation to a mobile collaborative application. The *Mobile Universal Identification (MUI)* identifies each mobile unit. This identifier allows the

SOMU Manager to make direct invocations to WS running on other mobile units. *WS Proxy* contains the information required to coordinate the invocation and the response of WS exposed by other mobile units. *WS Input* contains the invocation parameters to be sent by the WS Proxy when it invokes the remote WS. *WS Output* contains the results of a WS invocation.

The *Mobile Units Near Me* is the component in charge of discovering and recording the mobile units that are close to the current mobile device. This information is used to decide a good time to start an interaction with a specific mobile unit. This component uses a multicast protocol. It involves discovering the name, universal identification and the IP address of the mobile units belonging to the MANET.

Since Web services are typically accessed from different kinds of mobile computing devices, interoperability and personalization play an important role for universal access. The *Mobile Units Profile Manager* stores and manages information related to mobile units, such as the universal identification, hardware, software, and network capabilities. Web service can use this information to provide optimized contents for various clients. The two main components of the platform, i.e., the *μWebServer* and the *SOMU Manager,* are explained in the next two sub-sections.

4.1 *μ*WebServer

The *μWebServer* has the capability of exposing Web services and executing HTTP requests from Laptops and PDAs. The *listener* is responsible for managing client requests on a particular port. It performs validations and determines the most appropriate supporting components to carry out a request. The supporting components represent the implementation of a particular Internet protocol. The *μWebServer* implements supporting components for HTTP and SOAP.

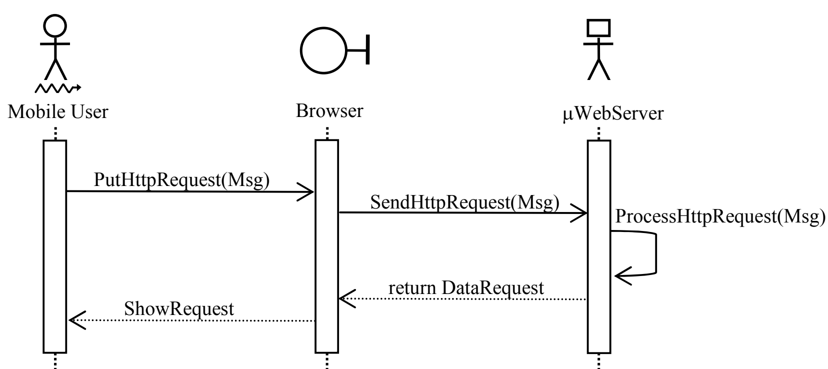
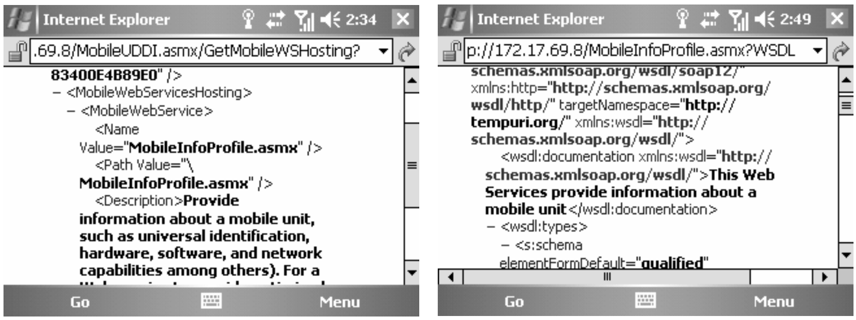


Fig. 3. Sequence diagram of result request service over HTTP

The HTTP component supports the processing of HTML, GIF and JPEG Web requests and GET and POST through SOAP components. As client requests are received, the required file is retrieved from local storage. Then, this file is converted into a stream of bytes and sent back to the mobile unit client. Figure 3 shows the sequence diagram to invoke Web services over HTTP GET operations.

Figure 4 (a) presents the results of invoking the "Mobile UDDI" Web service (included by default in SOMU), which provides information about all Web services hosted in a remote mobile unit. Figure 4 (b) presents the results of a similar invocation. In this latter case, the invoked remote Web service is the "Mobile Info Profile" Web service (also included by default), which informs the WSDL document related to it.



(a) (b)

Fig. 4. (a) List of Web services hosted in a remote mobile unit; (b) WSDL of a remote Web service

On the other hand, the SOAP component addresses the requirements of processing Web services remote invocations by clients. The current implementation supports GET, POST and SOAP action operations. Typically GET and POST operations are used for browser requests. These operations return an XML string representing the results. Meanwhile, SOAP actions are used to identify SOAP packets sent by applications using a particular Web service. Additionally, the SOAP component provides facilities to automatically generate WSDL (Web Service Definition Language) files from a requested Web service.

4.2 SOMU Manager

This component creates, stores and dispatches work items when a mobile collaborative application wants to invoke remote Web services. If the destination mobile unit is online, the SOMU manager picks up the work item and processes it, by creating a proxy client instance that interacts with the remote Web service. When the SOMU manager receives the results of the Web service invocation, it notifies to the mobile collaborative application and it delivers the results.

On the other hand, if the remote mobile unit hosting the Web services is not reachable, the mobile collaborative application switches to offline mode. Internally, the mobile collaborative application calls the manager to create a work item. The work item is stored in the *work items queue*. Periodically, the *mobile units near me* verifies if the destination mobile unit gets online. When the destination unit is online, the SOMU manager from the requester unit retrieves the work item from the queue.

Then, the manager sends an invocation of the remote Web service using the proxy functions. After processing the request, the remote Web service returns the results back to proxy client. Finally, the manager then returns the results to the mobile collaborative application.

4.3 SOMU Components Dynamic Interaction

In order to understand the SOMU platform functionality, Figure 5 shows the dynamics of the interactions between two mobile collaborative applications, when an application "A" requires a Web service exposed by a remote application "B". The first step of this interaction requires the application "A" make a local request to invoke the remote service from "B". "A" states this requirement through a work item which is created and stored by the SOMU manager (2nd step). Then, this manager asks to the *mobile units near me* component if the application "B" is in online mode and if "B" is within the "A" communication range. If the answer is negative, then the SOMU manager waits and retries until it gets a positive answer (3rd step).

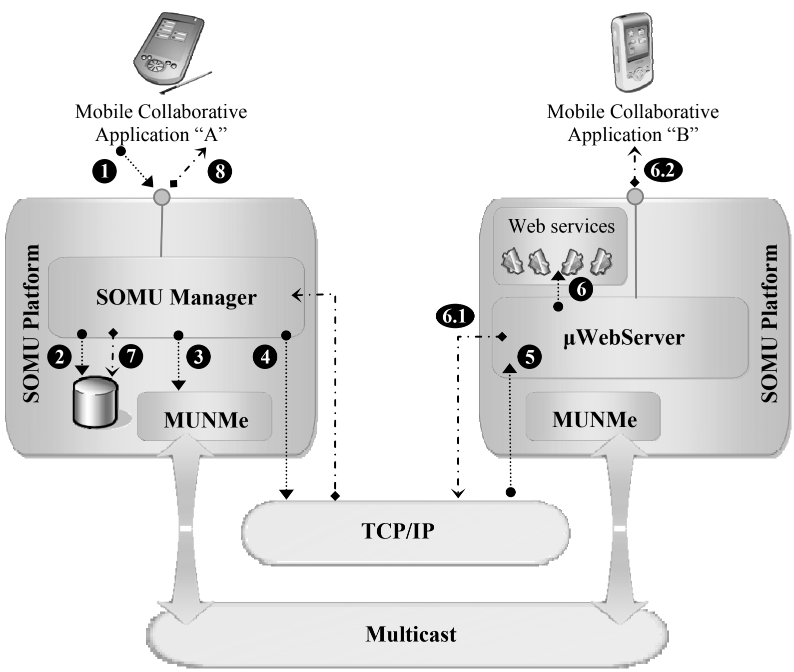


Fig. 5. Interactions among SOMU main components

When the mobile application "B" gets online and in the "A" communication range, the SOMU manager creates the proxy using reflection from the context information. Such information is in the WS Proxy field which is part of the work item. Then, the

SOMU manager invokes the remote service hosted in “B” (4th step). The invocation is received by the µWebServer (5th step). Since the request is a Web service invocation, the µWebServer SOAP component activates the corresponding Web service through reflection and it invokes the method implementing the service (6th step). The µWebServer returns the results to the mobile application “A” (6.1 step). If the application “B” is subscribed to receive the events related to a specific Web service, the µWebServer will send the corresponding notification (6.2 step). When the SOMU manager from “A” receives the results, it removes the work item from the queue (7th step). Finally, the SOMU manager from “A” notifies the mobile application “A”, indicating the work item with the specific ticket has finished its processing (8th step).

4.4 SOMU Implementation Aspects

SOMU was implemented in C# using the .NET Compact Framework; however, it can also be implemented using the J2ME SDK for mobile devices. The .NET platform was chosen since it offered rapid prototyping and a rich development environment including live debugging on emulators. The .NET libraries natively support XML manipulation, Web service description and reflection. This allows us to implement basic services for Web services description and discovery.

5 Application Scenarios

The following scenarios show how the actions taken by a mobile collaborative application are translated into the actions that occur within the Services Oriented Mobile Units. Two mobile collaborative applications which use the services provided by the platform were developed in order to test SOMU. These applications represent a proof-of-concept and they illustrate the feasibility of the proposed approach. One of them concerns one of the application scenarios mentioned in Section 1. They are briefly described below.

5.1 Mobile Electronic Meeting System

The implemented mobile electronic meeting system, called *Meeting Space*, is an interactive mobile computer-based system for supporting decision meeting processes. Like other Electronic Meeting Systems (EMS), the application goal is to support group members to be effective and make good decisions. The application was designed to be used by mobile users working online and offline. Provided the application running on each mobile unit is autonomous, uncoupled and independent of centralized components and networking infrastructure, users can meet in almost any place and carry out an ad-hoc work meeting (Fig. 6). The tool supports just some pre-meeting and meeting processes [4]. Specifically, it allows to: (1) create and share a meeting agenda, (2) specify, make private annotations and provide feedback about shared problems and solution ideas, (3) detect peers near the current device and generate notifications, and (4) share documents with peers.

During a pre-meeting, users can work alone in order to collect information and make private annotations about each item of the meeting agenda. When the *mobile users near*

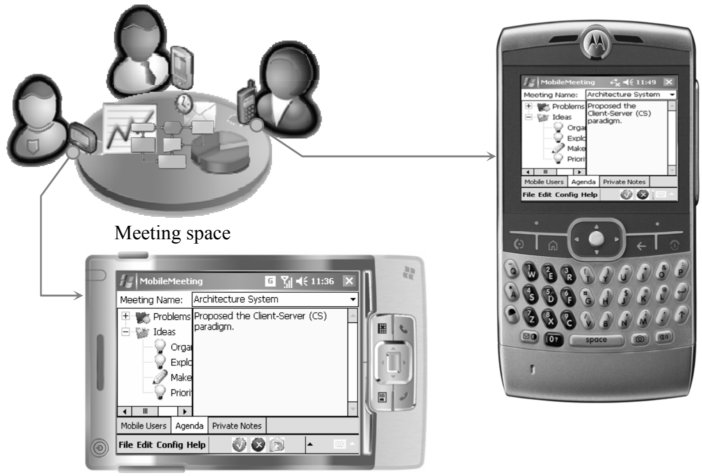


Fig. 6. Meeting Space Application

me component detects two or more users in the same communication range, it notifies the local SOMU manager. Then, this manager notifies the others SOMU managers running on the remote mobile units, which deploy a visual notification on the screen of the mobile computing devices. Thus, the application provides these persons the opportunity to hold an ad-hoc meeting to discuss in a face-to-face setting. For such discussion, the users can share documents and annotations by using the SOMU Web services and also specific Web services developed just for this application. The preliminary conclusions or results of the pre-meeting can be recorded in a shared file. Then, this file can be distributed among the mobile units by using the SOMU Web services.

Users can propose and share ideas (and annotations related to them) to discuss each item of the meeting agenda. Users can also provide feedback about the proposals and interact with other users in order to refine an idea, problem or any other item. The current application does not support rich computer-supported interaction mechanisms, such as full mediated discussion forums or brainstorming tools.

The meeting documents are registered and distributed to the corresponding members at the end of the meeting. Since the meeting place is almost any available place, the service-oriented solution proposed by this application becomes suitable to address the physical scenario constraints.

Figure 7 presents a possible sequence diagram of a process to show how the system supports an idea discussion. When a mobile user A proposes a public idea, the SOMU manager creates a work item. These public ideas can be delivered to other peer members as soon as possible. If a mobile user B is in A’s communication range, then A’s SOMU manager invokes a Web service from B in order to communicate the idea. When B’s µWebServer receives the proposal, it communicates the idea to the local application. The B’s application makes the idea available for the user to process it. If B rejects the idea, then such communication is received by the A’s application via the

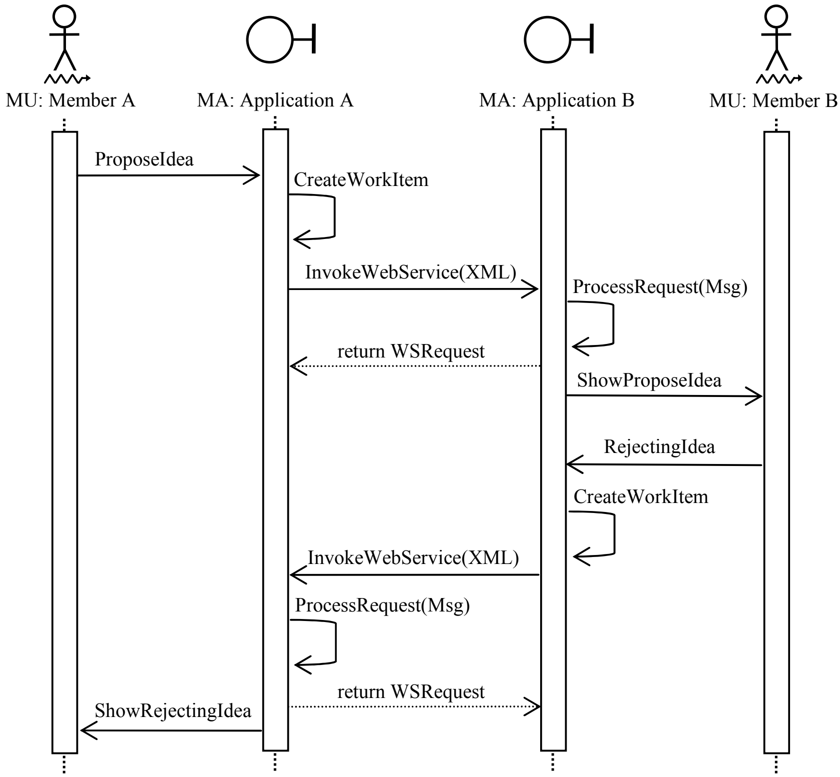


Fig. 7. Sequence diagram of typical activity of accepting/rejecting/refining ideas between two meeting members

SOMU manager. Afterwards, the A user can redefine the idea and submit it for consideration again. Thus, a new interaction cycle begins.

A next version of the *Meeting Space* application is planned to have support for voting. Users will be able to cast anonymous votes, a desirable feature in certain types of decision meetings. This feature will make the application valuable in physical settings otherwise unsuitable for these decision meetings.

5.2 Group Decision-Support System for Disasters in Urban Areas

Disasters affecting urban areas have shown the need to improve the group decision-making processes and the coordination of efforts done by organizations participating in disaster relief activities [6]. Typically, police is in charge of isolating and securing the affected area, firefighters are the initial responsible for protecting human life and physical infrastructure, medical personnel are responsible for healthcare of the affected people, and government authorities are responsible for coordinating the efforts of the participant organizations in order to reduce the impact of the extreme event on Society [6]. One key aspect to consider is that critical activities must be carried out in a short time period, such as ensuring the safety of the disaster area and conducting search and rescue procedures [25].

Initially, first responders deployed in the disaster scenario have to coordinate their efforts in order to support these activities. The developed application allows mobile first responders to access and distribute shared geographical information of the disaster area and the resources available to support the mitigation process. The information about the available resources is deployed on a map in order to get a visual identification of the resource allocation (Fig. 8). This information is divided in several layers. Each organization involved in the activities can update a specific layer. These information layers can be shared among mobile workers deployed in the disaster area and also among the disaster managers. Typically, a mobile worker uses a PDA or a Tablet PC.

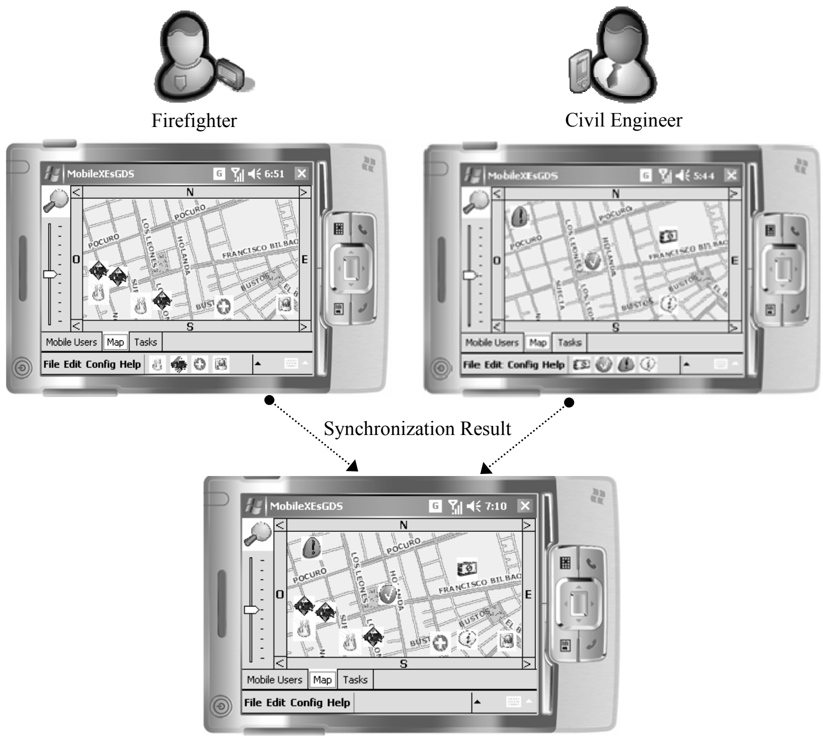


Fig. 8. Information Synchronization between a firefighter and a civil engineer

In order to illustrate how this application works, let us consider the following situation. A firefighter team needs to get updated information related to the stability of the physical infrastructure of an area, because they need to conduct search and rescue activities in such place. Therefore, the most direct way is to get an updated information layer from civil engineers evaluating the area. Two or more firefighter team members can use PDAs to get information from civil engineers, other partners and the command post.

If the communication in the disaster area is based on MANETs, then the firefighter team members need to be aware about the presence of civil engineers within their communication range. The *mobile units near me* component can notify these firefighters about such situation. Firefighters synchronize structural information from the civil engineer and get an updated view of the disaster area (Fig. 8). Thus, these first responders can make better decisions about where and when to conduct the search and rescue procedures. The decisions made and the results of the search and rescue activities are recorded in the firefighters' information layer. Now, a new update of the shared information is available.

This synchronization process uses not only the Web services provided by SOMU by default, but also other Web services created just for this application. One of these Web services is SyncXML that synchronizes to XML files following a policy similar to the one proposed by XMIDDLE [15]. This Web service is essential for the application because all basic information is represented in XML. In order to illustrate how SOMU components interact in this application, let us consider the same case described in Figure 8. In this case, Figure 9 presents a sequence diagram of the interactions between SOMU and application components.

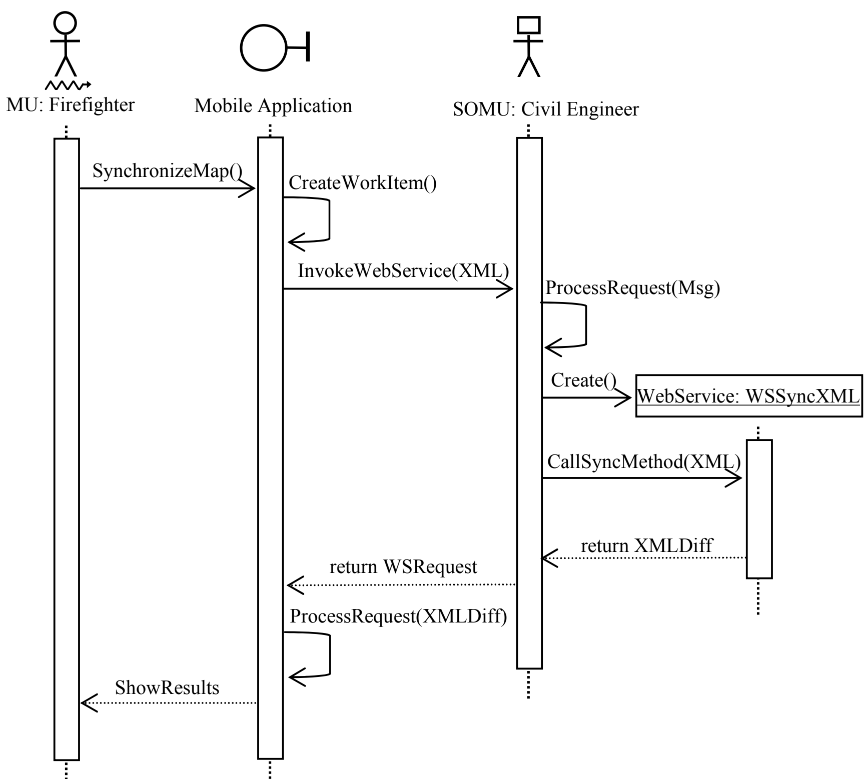


Fig. 9. Sequence diagram of synchronization structural information between a firefighter and a civil engineer

When a civil engineer gets in the communication range of the firefighter team, or vice versa, firefighters using PDAs are notified through a user event launched to the device screen. Therefore, a mobile unit used by a firefighter requests a synchronization operation in order to get updated information related to the physical infrastructure information layer. Then, the SOMU manager creates a work item and invokes a Web service exposed by the civil engineer's mobile unit, by indicating the version of the layer information the firefighters have. When, µWebServer in the civil engineer mobile unit receives the request, it launches the SyncXML service to process the request. Since the information the firefighter has is outdated, a local process is launched in the civil engineer mobile unit to retrieve the information updates from the local layer. Then, an XML file indicating the information updates is sent to the firefighter SOMU manager as response to the invocated Web service. The receiver mobile unit processes such information and shows it on the device screen.

6 Conclusions and Future Work

Most frameworks and platforms proposed to support collaborative activities of mobile workers use some type of centralized data or services. This centralization jeopardizes the application capabilities to support collaboration in ad-hoc communication settings. This paper presents a platform called SOMU (Service-Oriented Mobile Unit) intended to support the collaborative activities carried out by mobile workers in ad-hoc scenarios. Unlike the previous related works, SOMU proposes a fully decentralized architecture allowing mobile devices to act as autonomous units. The platform lets mobile computing devices expose and consume Web services in order to carry out an activity. Collaborative mobile applications developed on this middleware are then able to interact among them almost in any communication scenario. Availability of this tool is particularly relevant to support mobile collaboration when there is no stable communication support or no communication at all.

This middleware was implemented in C# using the functionality provided by the .NET Compact Framework. However, the same functionality could be implemented using J2ME. The type of implementation allows SOMU to run on a wide range of computing devices from PDAs to desktop PCs.

The platform provides a basic foundation for the development of mobile collaborative applications. This platform intends to increase the technical feasibility of solutions in the area and to reduce the development effort of MANET-based mobile collaborative applications. These issues have not been fully analyzed yet for the two developed applications, but the initial findings support these hypotheses.

Future work includes, in the short future, formal experimentation to study the possible contributions and limitations of SOMU and the consequences on the applications developed on it. Furthermore, the functionality provided by Web services will be tested in order to determine if the uncoupled interaction proposed by SOMU represents a limitation for mobile workers when collaborating in ad-hoc communication scenarios. As a second step, the functionality of SOMU will be extended to integrate (by default) P2P sessions management, standard WS discovery mechanisms (such as WS-Discovery), and enabled support for the new stack specification, such as WS-Security, WS-Trust, WS-Federation, WS-Addressing, WS-Routing and WS-Attachment.

Acknowledgments. This work was partially supported by Fondecyt (Chile), grant Nº: 1040952 and by MECESUP (Chile) Project Nº: UCH0109.

References

1. Adelstein, F., Gupta, K.S., Golden, R., Schwiebert, L.: Fundamentals of Mobile and Pervasive Computing. McGraw-Hill Publisher, (2005).
2. André, P., Antunes, P.: SaGISC: A Geo-Collaborative System. Proc. of the 10th International Workshop on Groupware (CRIWG), Springer, LNCS 3198, San Carlos Costa Rica, (2004) 175-191.
3. Bosneag, A.M., Brockmeyer, M.: GRACE: Enabling collaborations in wide-area distributed systems. Proc. of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE), Workshop on Distributed and Mobile Collaboration (DMC), IEEE CS Press, Linkoping University Sweden, (2005) 72-77.
4. Bostrom, R., Anson, R., Clawson, V.: Group Facilitation and Group Support Systems. In Group Support Systems: New Perspectives, Jessup and Valacich (Eds.), (1993) 146-168.
5. Buszko, D., Lee, W., Helal, A.: Decentralized Ad-Hoc Groupware API and Framework for Mobile Collaboration. Proc. of ACM International Conference on Supporting Group Work (GROUP), ACM Press, Colorado USA, (2001) 5-14.
6. Comfort, L.: Coordination in Complex Systems: Increasing Efficiency in Disaster Mitigation and Response. International Journal of Emergency Management 2(1), (2004) 62-80.
7. Gelernter, D.: Generative Communication in Linda. ACM Transactions on Programming Languages and Systems 7(1), (1985) 80-112.
8. Guerrero, L., Pino, J., Collazos, C., Inostroza, A., Ochoa, S.: Mobile Support for Collaborative Work. Proc. of the 10th International Workshop on Groupware (CRIWG), Springer, LNCS 3198, San Carlos Costa Rica, (2004) 363-375.
9. Handorean, R., Payton, J., Julien, C., Roman, G.: Coordination Middleware Supporting Rapid Deployment of Ad Hoc Mobile Systems. Proc. of the 23rd International Conference on Distributed Computing Systems (ICDCS), Workshop on Mobile Computing Middleware (MCM), IEEE CS Press, Rhode Island USA, (2003) 363-368.
10. Hauswirth, M., Podnar, I., Decaer, S.: On P2P Collaboration Infrastructures. Proc. of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE), Workshop on Distributed and Mobile Collaboration (DMC), IEEE CS Press, Linkoping University Sweden, (2005) 66-71.
11. Heinemann, A., Kangasharju, J., Lyardet, F., Mühlhäuser, M.: iClouds: Peer-to-Peer Information Sharing in Mobile Environments. Pro of the 9th International Euro-Par Conference (Euro-Par), Springer, LNCS 2790, Klagenfurt Austria, (2003) 1038-1045.
12. Hirsch, F., Kemp, J., Ilkka, J.: Mobile Web Services: Architecture and Implementation. Nokia Research Center. John Wiley & Sons Publisher, (2006).
13. JXTA Project, URL: http://www.jxta.org. (2003).
14. Marquez, J., Navarro, L.: Autonomous and Self-sufficient Groups: Ad Hoc Collaborative Environments. Proc. of the 11th International Workshop on Groupware (CRIWG), Springer, LNCS 3706, Porto do Galinhas Brasil, (2005) 57-72.

15. Mascolo, C., Capra, L., Zachariadis, S., Emmerich, W.: XMIDDLE: A Data-Sharing Middleware for Mobile Computing. Journal on Personal and Wireless Communications 21(1), (2002) 77-103.
16. Meissner, A., Musunoor, S.: Group Integrity Management Support for Mobile Ad-Hoc Communities. Proc. of the 4th International Middleware Conference, Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC), ACM Press, Rio de Janeiro Brazil, (2003) 53-59.
17. Menchaca-Mendez, R., Gutierrez-Arias, E., Favela, J.: Opportunistic Interaction in P2P Ubiquitous Environments. Proc. of the 10th International Workshop on Groupware (CRIWG), Springer, LNCS 3198, San Carlos Costa Rica, (2004) 349-362.
18. Muñoz, M.A., Rodriguez, M., Favela, J., Martinez-Garcia, A.I., Gonzalez, V.M.: Context-aware mobile communication in hospitals. IEEE Computer, 36(9). (2003) 38-46.
19. Nemlekar, M.: Scalable Distributed Tuplespaces. MSc. Thesis. Department of Electrical and Computer Engineering, North Carolina State University, Chapter 5, (2001).
20. Neyem, A., Ochoa, S., Guerrero, L., Pino, J.: Sharing Information Resources in Mobile Ad-hoc Networks. Proc. of the 11th International Workshop on Groupware (CRIWG), Springer, LNCS 3706, Porto do Galinhas Brasil, (2005) 351-358.
21. Procopio, M.J.: YCab.NET: Decentralized Collaboration Groupware for Mobile Devices using the MS .NET Framework. Master's of Science Thesis, University of Florida. (2002). Available at http://etd.fcla.edu/UF/UFE1000155/procopio_m.pdf
22. Sen, R., Handorean, R., Roman, G-C., Gill, C.: Service Oriented Computing Imperatives in Ad Hoc Wireless Settings (Book Chapter), Service-Oriented Software System Engineering: Challenges And Practices, Stojanovic and Dahanayake (Eds.), Idea Group Publishing, Hershey, USA, (2005) 247-269.
23. Stojmenovic, I., Wu, J.: Ad-hoc Networks. IEEE Computer, 37(2), (2004) 9-74.
24. Tschudin, C., Lundgren, H., Nordström, E.: Embedding MANETs in the Real World. Proc. of the 8th International Conference on Personal Wireless Communications (PWC), Springer, LNCS 2775, Venice Italy, (2003) 578-589.
25. Turoff, M.: Past and Future Emergency Response Information Systems. Communications of the ACM, April, (2002) 29-32.
26. Zurita, G., Baloian, N.: Handheld Electronic Meeting Support. Proc. of the 11th International Workshop on Groupware (CRIWG), Springer, LNCS 3706, Porto do Galinhas Brasil, (2005) 341-350.