A Service-oriented Mobile Social Networking Platform for Disaster Situations

Xiping Hu[1], Victor C.M. Leung[1], Weichang Du[2], Boon-Chong Seet[3] and Panos Nasiopoulos[1]
[1] Dept. Electrical and Computer Engineering, The University of British Columbia
[2] Faculty of Computer Science, University of New Brunswick
[3] Dept. Electrical and Electronic Engineering, Auckland University of Technology
Email: {xipingh, vleung}@ece.ubc.ca, wdu@unb.ca, boon-chong.seet@aut.ac.nz, panos@icics.ubc.ca

Abstract

This paper proposes a novel service-oriented mobile social networking platform called MS2A for disaster situations. MS2A consists of two layers: service layer and application layer. The service layer is implemented based on the service-oriented architecture (SOA) to ease application developments. By extending and composing the web services at this layer, application developers can efficiently and flexibly develop different new functions or applications for mobile disaster rescue and recovery applications. The application layer is oriented to mobile users and it integrates with functions of mobile social networks. With dynamic and automatic service collaboration support, it enables people to easily collaborate and help each other through their mobile devices in disaster situations. The results of our experiments and demonstration examples given in the paper show the feasibility and desired functionality of MS2A for relieving disaster situations with mobile devices.

1. Introduction

When a disaster such as an earthquake strikes, the electricity and telecommunication networks may be damaged so that reliable communications and Internet access are no longer available, and as a result people's ability to seek help as well as emergency management coordination processes may suffer considerable delays. One of the important requirements in such situations is to support people to communicate in an infrastructure-less way [1], as it may take too much time to install new communication equipment and restore the damaged infrastructures. With the advancements of wireless communication and information system technologies, new forms of mobile technologies can be effectively utilized in disaster situations to address the above requirement. For example, advanced mobile devices such as smart phones now have the capability of forming opportunistic networks in an ad-hoc manner through WiFi Direct or Bluetooth connectivity [2], which can provide an efficient and effective means of supporting rescue and recovery operations immediately after disasters [3,4]. However, due to the variety of disaster situations, e.g., earthquake, hurricane, and tsunamis, and the potentially large number of individuals affected in a disaster, many diverse service requirements for different individuals may co-exist in a specific disaster situation. For instance, some people may be more concerned about accurately finding their current positions, some may be more concerned about locating the nearest rescue center, while some may have urgent needs to interact with the nearest rescue workers. Thus, a single ad-hoc network based mobile application or an existing wireless communication system for rescuers based on a single radio technology may not be able to meet a significant portion of the potential service requirements of different disaster situations [5, 6].

Further, as the situations in a disaster may change very quickly and become highly unpredictable, it is almost impossible to forecast every possible scenario and the related service requirements before a disaster, or even after it has happened. One of the effective ways to address this challenge is to enable people to directly collaborate with each other through mobile devices (e.g., smart phones) that they carry with them every day [7]. Nevertheless, today's mobile devices may run different operating systems, support different functions and follow different interaction standards. Although people can interact and collaborate with each other through some popular Internet-based social network applications like Facebook and Twitter, in disaster situations reliable Internet access may not be available. Consequently, it is of interest to develop an effective platform or system that can leverage the advantages of social networks and enable them to work independently upon wireless ad-hoc networks, so as to dynamically enable a variety of rescue tasks and collaborative applications in disasters.

A disaster relief operation mainly consists of three stages: 1. preparation stage before the disaster happens; 2. immediate response stage just after the disaster has happened; 3. recovery stage after the disaster was over

1530-1605/12 $26.00 © 2012 IEEE
DOI 10.1109/HICSS.2013.59

[8, 9]. In this paper, we propose a novel system design called MS2A that incorporates a service-oriented architecture (SOA) to enable an application development environment capable of utilizing mobile social networks (MSNs) for applications supporting disaster rescue and recovery operations. Compared to existing SOA-based systems for disaster situations, MS2A fully supports SOA based mobile applications working independently in popular mobile platforms (e.g., Android) in an ad-hoc manner with low resource overhead in mobile devices. MS2A aims to support application developers to efficiently develop diverse and useful applications in anticipation of mobile disaster rescue and recovery operations for the first stage, and to enable people in disasters to effectively collaborate with others in their vicinity through their mobile devices for the second stage.

The organization of the rest of the paper is as follows. Section 2 reviews the background of the related techniques of MS2A. Section 3 introduces the overall design of MS2A, presents its unique services and functions, and discusses why it can meet the challenges in disaster situations mentioned above. Section 4 analyzes the development challenges and implementation strategies of MS2A. Section 5 shows application examples to demonstrate and evaluate the proposed MS2A system. Section 6 reviews other systems proposed for disaster situations and briefly compare them with MS2A. Section 7 concludes the paper with remarks on the prospects of MS2A for future applications.

2. Background

MSNs are social networks in which participants with similar interests and/or objectives interact with one another through their mobile devices [10]. Different from conventional social networks in which people interact over the Internet [11], MSNs can also work over opportunistic networks, where each node can act as a host, a router or a gateway and connect with other nodes in an ad-hoc manner, without possessing or acquiring any knowledge about the network topology [12, 13]. Therefore MSNs are attractive for supporting interactions and collaborations between people in disaster situations.

SOA is a set of specifications and methodologies for designing and developing software applications/services in the form of interoperable service components [14]. Services in SOA are loosely coupled, have unified specifications for service interactions but do not depend on system-specific languages. This flexible architecture can effectively support dynamic service composition and customization for individual applications or users. Thus, application developers can flexibly and efficiently combine different services in SOA to generate many kinds of new applications according to their diverse service requirements for disaster rescue and recovery operations. Even though the developed applications are different, they still have the capabilities to collaborate (i.e., sharing data) with each other across different operating systems in their run-time. This is especially useful in disasters situations, since mobile devices are normally isolated from the Internet and lack sufficient functions on their own in such situations.

However, all of the current SOA based solutions for disaster situations employ SOA clients in mobile devices [15-17] that access web services deployed on specific servers via the Internet; thus SOA based mobile applications are not capable of running in the mobile devices independently and collaborating with each other directly over opportunistic networks. The possibility of damaged infrastructure in disaster situations diminishes the viability of this approach as it has a high network overhead and relies on centralized facilities. With the improvements of computing and communication abilities of mobile devices, the conditions are becoming mature to support the deployment of light weigh SOA servers and web services on the mobile platforms. For instance, the computation power of a smart phone with NVidia Tegra 3 CPU is more powerful than a laptop with Intel T7200 CPU, which was popularly used 5 years ago, while its power consumption is only at most 11% of the later [18]; and most smart phones are equipped with 802.11n radios supporting data rates up to 300Mbps [19].

3. System design

The system architecture of MS2A is shown in Figure 1. MS2A is an extensible and configurable mobile disaster management system that is based on SOA. As mentioned in Section 1, MS2A aims to provide a universal platform to support the diverse service requirements of disaster rescue and recovery operations. It consists of two components: the application layer and the service layer, both of which can run on the same mobile device simultaneously. The application layer is oriented to end users who can directly use the MS2A system in their mobile devices through MS2A's user interface; while the service layer is transparent to end users and oriented towards supporting efficient development of applications for mobile disaster rescue and recovery operations. Using the service-oriented programming model, software developers can easily and efficiently implement diverse mobile applications for disaster scenarios.

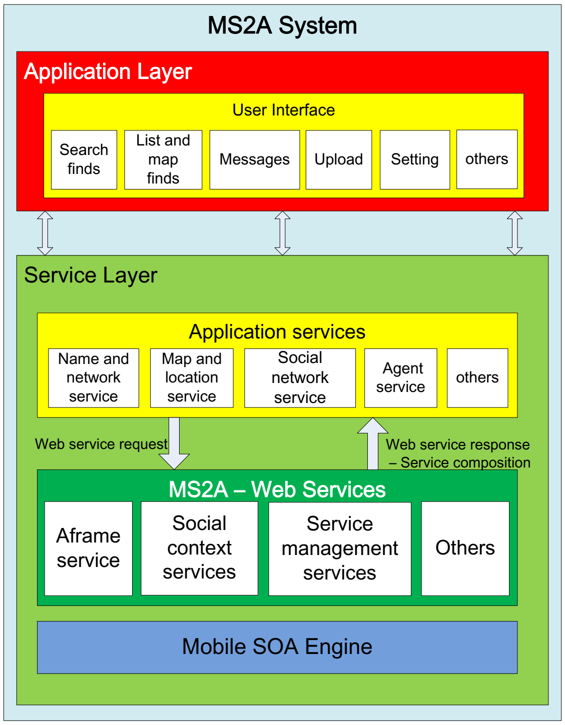


Figure 1. The MS2A system

3.1. Service layer

As shown in Figure 1, the service layer consists of two sub-layers: web service layer and application service layer. Existing web services are located at the web service layer and available to application developers. These web services are enabled by the mobile SOA engine, which is the SOA server running in the mobile devices. More details about the implementation of the web services and interaction between the web services and mobile SOA engine will be discussed later in Section 4. The application service layer is found on top of the web service layer, and provides sets of web services to specific applications.

A. Web service layer

As discussed in Section 1, due to the different types of disasters, diverse service requirements of rescue and recovery operations, and quickly changing environments in disaster situations, it is crucial to provide an application development environment that enables mobile applications to operate over networks formed in an ad-hoc manner, and supports collaborations between these applications in their run-time. The web service layer is designed to provide basic service components that developers can incorporate into applications for disaster situations. All the services at this layer are SOA based web services. Thus application developers can take full advantage of SOA to efficiently and flexibly extend the web services, or integrate the web services at this layer to generate different application services. Moreover, developers can also design mechanisms or algorithms for service compositions at this layer, so that multiple applications developed on MS2A can automatically and intelligently access, use or collaborate with the running web services shared among the mobile devices at run-time, which is especially useful for dynamically and quickly changing environments in disaster situations. In the current version of MS2A, there are mainly three types of web services at this layer: Aframe [20, 21] service, social context services, and service management services.

Aframe service

Aframe is an agent-based application programming framework for mobile wireless ad-hoc network, which we have designed and implemented previously. By integrating AmbientTalk (a new programming language for mobile ad-hoc networks) [22] and software agent technique in a multi-layer framework, Aframe can simplify and hide the complexity of handling different connectivity status of mobile wireless ad-hoc networks, and various services requirements. It supports collaborations among multiple agents working in multiple devices simultaneously, such as supporting multiple mobile agents to dynamically and self-adaptively execute different applications around wireless ad-hoc networks with the help from resident agents. For the MS2A system, we implement the whole Aframe framework as a web service at this layer. Further, as Aframe also consists of several services of its own, these Aframe services can be seen as sub-services at this web service layer of MS2A. There are mainly two kinds of services inside Aframe: application specific services and framework services. Application specific services can be developed and extended by application developers according to their specific service requirements, while framework services are generic services as follow:

ID name service: It generates a universally unique identifier (UUID) as ID name for each mobile device.

Data fetching and storage service: It provides shared and non-shared data services to mobile agents through resident service agents.

Network status service: It provides real-time information of currently connected mobile devices in the local mobile wireless ad-hoc networks.

Migration service: Using this service, mobile agents can migrate around the mobile devices in the underlying mobile wireless networks.

Current time service: It provides local timing information in a mobile device.

Deployment service: Using this service, resources and specific services can be deployed and installed around the local mobile devices dynamically by mobile agents.

Moreover, in order to support the functionality of MS2A, we also implement two application specific services inside Aframe: Location service, which is used to get the location information of mobile devices; and Map service that is used to invoke the Google Map application program interfaces (APIs) in mobile device and sign the location information in the digital map. In addition, since the current mobile version of Google Map supports pre-storing the digital map and offline map wraps, which enables it to work independently from the Internet, so can the Map service of MS2A. Further, taking advantage of SOA, the Map service can collaborate with each other among the mobile devices in their run-time, e.g., to support sharing and compositing the map wraps of Google Map when some digital map information is available in one mobile device but not another.

Social context services

This type of service makes use of the open APIs of popular MSN applications like Facebook and Google+, and encapsulates them as web services in MS2A, so as to leverage their functions to MS2A and enable them to work independently from the Internet. Application developers can extend, invoke or composite this type of service with other web services at this layer. For example, developers can composite these social services with network services of Aframe, so that the resulting service can support interactions between people through conventional MSN application over wireless ad-hoc networks. In other words, it leverages the advantages of the MSN applications in MS2A, since most people are familiar with their operations, and they have already provided mature and sufficient ways of social interaction. The current version of MS2A mainly consists of three services as follows:

Social contacts service: Normally, the popular MSN applications can synchronize their users' contact information to the address books of the smart phones. This service makes use of the related open APIs of the MSN applications and smart phones, and integrates all the contact information as a service that can provide the contact information of MSN users.

Social interaction service: It is used to support the information interaction, such as sending XMPP (Extensible Messaging and Presence Protocol) based unicast messages between the users.

Information publication service: It is used to support publication of information through sending multicast messages to the nearby mobile devices connected to the local wireless networks.

Service management services

This type of service is used to manage all the available web services of MS2A. Application developers can also design new mechanisms or extensions based on these web services before they develop concrete applications. The following services are contemplated:

Service list service: It is used to fetch the current available web services in the local mobile device and the underlying local wireless networks.

Registration service: It is used to register new web services that are extended to the MS2A system during run-time.

Monitor and control service: It is used to monitor the status of all the web services of MS2A.

Service transfer and adaption service: It is used to transfer and deploy the web services of MS2A among the mobile devices, and support the replacement of the web services in their run-time.

Schedule and coordinate service: It is used to configure the current web services and support the collaboration among the web services of MS2A.

B. Application service layer

Every service at this layer is implemented through invoking or compositing the web services at the lower layer as mentioned above. This layer is oriented towards the realization of concrete applications, and every service at this layer can work independently. Application developers only need to develop the user interface with the related application service at this layer, through which the end users of MS2A can directly use the corresponding function. Currently, there are mainly four application services implemented at this layer to support the functionality of MS2A: name and network service, map and location service, social network service, and software agent service. As all of these services are related to the application functions of MS2A, thus more details about them will be introduced in the next section.

3.2. Application layer

The application layer of MS2A directly serves the needs of the end users in disaster situations. The current version is designed to assist rescuers in the search of victims and key personnel and resources (e.g., the nearest doctor, the nearest medical supply). It could also enable the victims to find the nearest persons who may be able to help. In addition, as discussed above, it integrates with popular MSN applications like Google+, thus people in disasters could also interact and collaborate with each other through sending messages over the local wireless ad-hoc networks when the communication infrastructure is not available. Moreover, since each function at this layer is implemented through the compositing of the underlying web services, thus users could also collaborate with each other through service collaboration. One example about this will be demonstrated later in Section 5. Application developers could also implement additional functions here according to their practical requirements. The user interface of the current version of MS2A is shown in Figure 2 and mainly consists of the following modules:

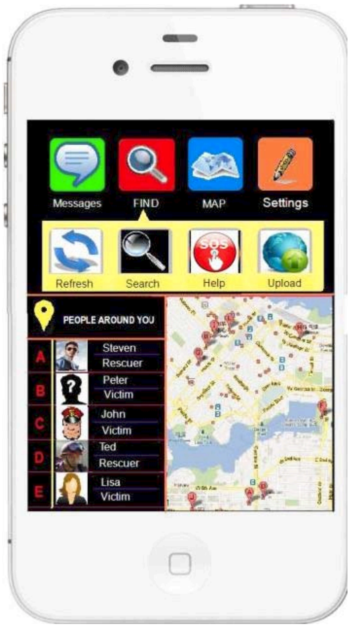


Figure 2. The user interface of the MS2A

Search finds: This function is based on the network and name service, which is implemented through the compositing of web services (i.e., ID name service, Network status service, Data fetching and storage service, etc.) at the lower layer. It can help the users to find out using the local wireless ad-hoc networks the other people who are currently nearby. During a search mission, when a new victim or a rescuer is discovered, the mobile devices will record his identity and related information such as position information and generate a time stamp.

List and map finds: This function is mainly based on the map and location service. It provides a list of all the found persons currently in the users' nearby places, and displays them both as a clickable list and as clickable icons on the digital map, e.g., Google Map. As shown in Figure 2, the information also includes the found persons' identities, brief descriptions and related small size camera photos. In addition, as the positions of users may change dynamically or they may even move out of the communication range, users can choose to manually or automatically refresh the positions information of the found persons, so as to figure out their latest positions or track their movements. For example, the victims may be concerned about where the rescuers are currently located, and whether the rescuers are coming towards them or not.

Message: This function is based on the social network service mentioned above. Similar to the use of conventional social networking applications, users can interact with the nearby people when they are found, sending text messages with photos, publishing information among people whose mobile devices are connected to the local wireless ad-hoc networks or Internet, so as to collaborate with each other in disasters. For instance, users can find out the nearest rescue center with the help from other users through publishing the related questions to the local MSN. In addition, users of MS2A can also predefine a message like "SOS" that would be sent out automatically through the software agent service when the user has not moved for a predefined time period, as determined by the sensor in the mobile device. This is especially useful for victims in disasters, as they may have lost consciousness after the occurrence of the disaster.

Upload: This function is also based on the network and name service. The data about disaster rescue of MS2A (i.e., the identities and time stamp of the found people, etc.) stored on the local mobile device can be uploaded periodically either manually or automatically to the remote disaster control center using Hypertext Transfer Protocol (HTTP) via the Internet when the communication infrastructure is available.

4. Implementation strategies

As indicated in Section 3, in order to implement the MS2A system, the major task is to implement its web service layer. Currently, the web services enabling most SOA based mobile applications are deployed on centralized servers while the mobile devices only run the SOA clients, which normally interact with the web services using Simple Object Access Protocol (SOAP). However, the web service interactions among the mobile devices need to rely on the central servers, which may not be reachable in a disaster situation. The loss of connectivity to the central servers may constrain the collaboration abilities among the mobile devices. Therefore, we adopt a new SOA design style called REpresentational State Transfer (REST) [23] to implement the web service layer of MS2A. Comparing to SOAP, the REST based SOA has several advantages in mobile platforms, including being: (i) more light weight with lower hardware requirements than SOAP based SOA servers that could hardly run on the mobile devices; and (ii) consuming about 56% less power than SOAP based SOA [24], which is very crucial in disaster scenarios as the mobile phones normally have limited battery power and it may not be possible to recharge them in many situations. In the rest of this section, we introduce the implementation architecture and process flow of the mobile SOA and related web services in the MS2A system.

4.1. The overall implementation of mobile SOA in MS2A

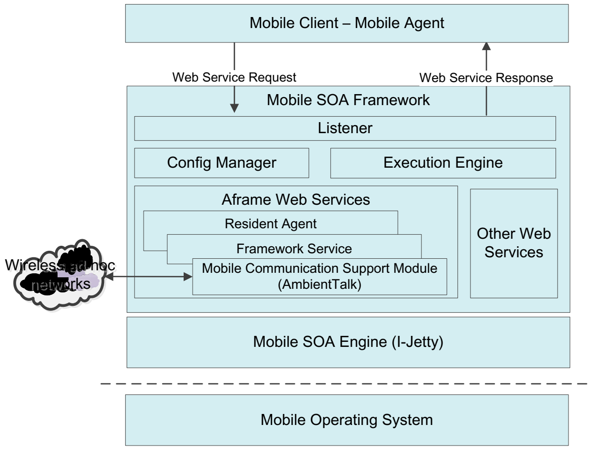


Figure 3. The overall implementation architecture of mobile SOA in MS2A

The overall implementation architecture of mobile SOA in MS2A is shown in Figure 3. Efforts to migrate conventional SOA frameworks like Jersey or AXIS to current mobile operating systems were unsuccessful due to the limited Java libraries offered by such systems. Thus we adopt a light weight SOA server for MS2A. Even though its ability in concurrency service interaction is not as high as traditional Internet-based SOA servers, it should be able to meet most of the service requirements in disaster scenarios. There are already some lightweight web servers available in mobile operating systems, such as the kWS Android Web Server (supports HTTP 1.0 and static web pages) [25], PAW server for Android (supports PHP plugin) [26], and I-Jetty open source web container based on Android (supports Java Servlet) [27]. Based on the Java service implementation we are using, we use I-Jetty to implement the mobile SOA engine of MS2A, which is at the bottom of the architecture. On top of the mobile SOA engine are the Java class libraries, which mainly consist of the encapsulated Aframe components and other Java class libraries, such as Aframe's communication module. This module is implemented in AmbientTalk to support wireless ad-hoc network communications. The resident agent module supports the service invocation with mobile agents. All of the web services based on the REST style are deployed in the mobile SOA framework and supported by the underlying Java class libraries. Finally, external to the mobile SOA are the mobile clients implemented to enable specific applications. MS2A uses the mobile agent implemented in Aframe as the default mobile client. However, application developers can also develop other types of mobile clients that are not necessarily agent based. One requirement here is that the implementation of a mobile client needs to correspond to the resource operation of the underlying REST style based web services, and their mapping relation is implemented through the configuration files in the mobile SOA framework.

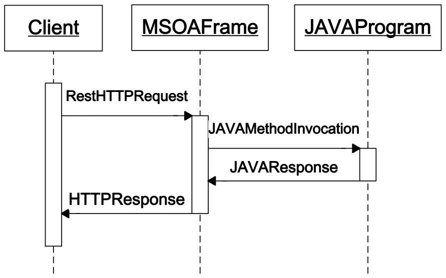


Figure 4. The process flow of mobile SOA in MS2A

The process flow of mobile SOA in MS2A is as shown in Figure 4. During run-time, the mobile SOA framework will firstly receive the web service request from the mobile client, and then analyze the request Uniform Resource Locator (URL) and the related parameters encapsulated by HTTP, so as to determine the specific Java class to invoke the corresponding web services based on the configuration files. Finally, after the operation of the related web services, the mobile SOA framework returns the results to the client in the form of REST style data through HTTP. One advantage of this architecture design is that the application developers do not need to be concerned with issues of mapping relation about specific URL to corresponding service resources, but can focus on the development of the application itself.

4.2. Implementation of the Aframe web service

The Aframe service plays an important role in the web service of MS2A. It not only supports the mobile devices forming opportunistic networks in an ad-hoc manner, but also enables the mobile applications developed on MS2A to work intelligently upon such networks. Therefore in this section we focus on the introduction of the Aframe service implementation in MS2A. The implementation methodologies of other web services are similar to it.

Different from the traditional SOA design concept like SOAP, which is process-oriented, the REST based SOA is resource-oriented, such that every URL in HTTP corresponds to a specific resource type, and the operations to the resource like *get, create, update* and *delete* correspond to the method *GET, POST, PUT* and *DELETE* of HTTP in REST. Therefore, as a first step the related URL of every key resource of Aframe in MS2A needs to be specified. For instance, if we consider the mobile wireless ad-hoc network communication support module – AmbientTalk of Aframe to be a resource, then we can assign an URL to this resource like */rest/ambienttalk*, and it will create a new ad-hoc network based communication resource to the mobile client when the client sent a POST request to this URL. In the second stage, the components of Aframe need to be encapsulated in terms of the Java operations, and finally a configuration file needs to be designed to map the implementation method from URL to Java operations. An example of a configuration file for the AmbientTalk module is shown in Table 1.

Table 1. Configuration file for AmbientTalk of Aframe in MS2A

URL	HTTP method	Java class	Operation
/rest/ambienttalk	GET	org.ubc.aframe. AmbientTalk	get AmbientTalk ()
/rest/ambienttalk	POST	org.ubc.aframe. AmbientTalk	create AmbientTalk ()
/rest/ambienttalk	PUT	org.ubc.aframe. AmbientTalk	Update AmbientTalk ()
/rest/ambienttalk	DELETE	org.ubc.aframe. AmbientTalk	Remove AmbientTalk ()

Here we use an example to illustrate how the service of Aframe works in the mobile SOA framework. As mentioned in Section 3.1, a location service exists as an application specific service in Aframe. In order to implement this service as a web service in MS2A, firstly we can configure the resource type of this service as Global Positioning System (GPS) information with corresponding URL: /rest/GPS, and then encapsulate the get method of the GPS information in the executeService() method of GPSService class, and register this method as the type of application specific service in the resident agent.

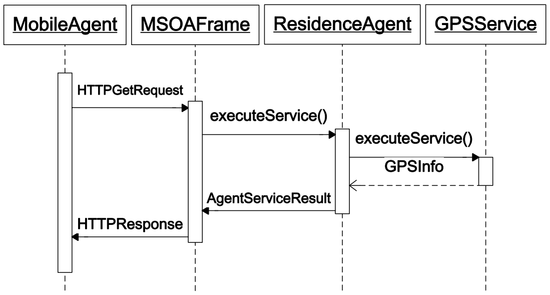


Figure 5. The process flow of location service

The process flow of this service is shown in Figure 5. Firstly, the mobile agent as a client will send the Get request to URL: /rest/GPS, and then the mobile SOA framework will invoke the executeService() method in the resident agent based on the configuration file when it receives the request. Secondly, the resident agent will find out the corresponding service - GPSService based on the parameters of the request, and then invoke this service and return the result to the mobile SOA framework. Finally, the framework will encapsulate the result as a HTTP response and return it to the mobile agent.

Moreover, in the current version (e.g., v4.0) of Android, each Android device can serve as a hotspot (communication server) and support 8 other devices connected to it through WiFi simultaneously, but in this case the node acting as a hotspot could not be a slave node and connect to other hotspots through WiFi again. Therefore in MS2A, we use both WiFi and Bluetooth techniques to support the forming of wireless ad-hoc networks, so as to enable each node to act as a communication server and client at the same time.

5. Application example and evaluation

In this section, we present an experiment that was conducted to prove the concept of MS2A and evaluate its ability by sending MSN messages over wireless ad-hoc networks. We also use an example to demonstrate the functionality of dynamic service collaboration of mobile SOA in MS2A.

5.1. Experiment

The experiment consists of four persons each carrying an Android device (three Android phones and one Android tablet) equipped with 802.11n wireless network module and running MS2A. We use one of the Android devices to act as a WiFi hotspot and let the others connect to it through WiFi. It was found that the initialization time of MS2A in all the devices and the time to form a local wireless ad-hoc network took only less than two minutes. The mobile devices normally have an ideal WiFi communication range up to 500 meters at a data rate of 15.5 Mbps or higher when there is no obstacle [19]. Each person carries a mobile device denoted as node M_x (M_2 is the hotspot in this experiment) and moves in an area of about 400*400 m^2 as shown in the map in Figure 6. Because of the experimental constraints in mobile devices, we only tested the stability of MS2A's message interaction. As shown in Figure 6 and Table 2, there are eight predefined positions, and each person sends MSN messages every 50s through MS2A to 2 – 3 destination nodes when he/she moves from one position to another at a normal walking speed. The duration of this experiment last for 15 minutes, and it was found that all sent messages were received successfully.

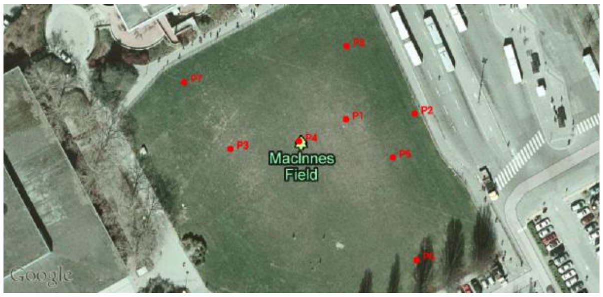


Figure 6. The map area of the experiment

Table 2. Experimental results

Node	Position	Message destination	Success
M_1	P_1, P_3	M_2, M_3, M_4	Yes
M_2	P_2, P_4	M_1, M_3	Yes
M_3	P_7, P_8	M_2, M_4	Yes
M_4	P_5, P_6	M_1, M_2, M_3	Yes

5.2. Demonstration example

We can easily see that the location service is very crucial in disaster situations as people are often concerned about their positions in such situations. Obtaining accurate location information normally needs GPS support. However, not all mobile devices, e.g., the Android tablet used in the above experiment, are equipped with GPS receivers. As discussed above, different from the former SOA based mobile applications for disaster situations, MS2A fully supports the SOA based mobile applications working independently without central servers while enabling collaborations between devices during run-time. To illustrate this capability, we have designed a demonstration application based on MS2A that enables the mobile users to share the GPS service.

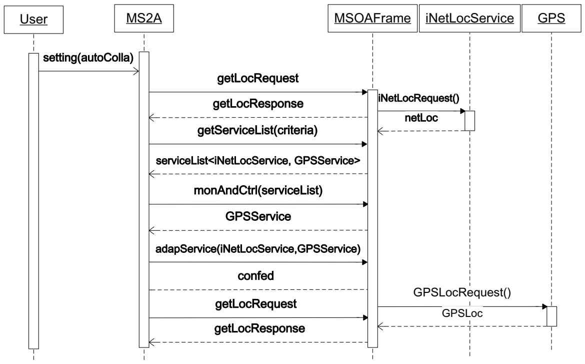
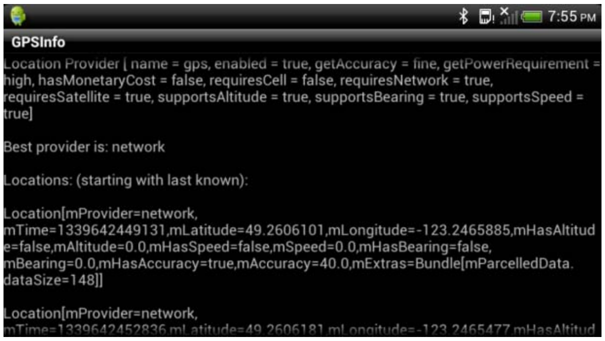


Figure 7. Process flow of dynamic web service collaboration for location service



```
Location Provider [ name = gps, enabled = true, getAccuracy = fine, getPowerRequirement =
high, hasMonetaryCost = false, requiresCell = false, requiresNetwork = true,
requiresSatellite = true, supportsAltitude = true, supportsBearing = true, supportsSpeed =
true]

Best provider is: network

Locations: (starting with last known):

Location[mProvider=network,
mTime=1339642449131,mLatitude=49.2606101,mLongitude=-123.2465885,mHasAltitud
e=false,mAltitude=0.0,mHasSpeed=false,mSpeed=0.0,mHasBearing=false,
mBearing=0.0,mHasAccuracy=true,mAccuracy=40.0,mExtras=Bundle[mParcelledData.
dataSize=148]]

Location[mProvider=network,
mTime=1339642452836,mLatitude=49.2606181,mLongitude=-123.2465477,mHasAltitud
```

Figure 8. Results of the service collaboration

In the example, we use two mobile devices: one equipped with a GPS receiver and the related service – GPS service (sub-service of the location service in MS2A); the other one does not have the GPS module but contain the location service of MS2A, and both of them are equipped with the Bluetooth v4.0 module offering up to 60 meters communication range [28]. When both of them run the MS2A and connect together through Bluetooth, with the help of service management services in MS2A, these two devices can automatically share the GPS service. Therefore, the device that does not have the GPS module can also obtain its location information with considerable accuracy. The corresponding process flow is shown in Figure 7. Finally, from Figure 8, it can be seen that the device finally obtain its location information with a precision of 40 meters (note the "gps status is true" but provider is network in Figure 8) from the GPS service in another device with the help of service collaboration of MS2A.

6. Related work

There has been much research work done on SOA for disaster situations. The use of grid computing was proposed [29] for the implementation of SOA for disaster monitoring. Using the concept of SOA, distributed service integration for disaster monitoring sensor systems was presented [30]. An SOA based earthquake disaster mitigation system [31] was designed. Here we focus on two representative cases.

MobiSoC [15] is a middleware that provides a common platform for rapid development and deployment of mobile social computing applications. Through capturing the social state from physical communities, and mining previously unknown patterns from emergent geo-social data with social state, a mechanism is designed to share the data of social state among the mobile devices to support real-time MSN applications. As the architecture of MobiSoC is also based on SOA, it supports evolution by providing modularity, extensibility, and language independence as well. Nevertheless, since the architecture design of MobiSoC is based on SOAP style SOA, as mentioned above, its service interactions rely on a SOA central server with relatively high network overhead that is hardly affordable in mobile wireless ad-hoc network environments. Thus, MobiSoC is not expected to work efficiently and be widely used in the disaster rescue scenarios.

WIPER [32] is an emergency and disaster information system designed and implemented based on dynamic data driven application system (DDDAS) concepts [33]. Through using enhanced situational awareness with high resolution data in the physical vicinity of a communication, WIPER can dynamically respond to streaming real-time data and use them with its agent-based simulation systems, so as to classify and predict the potential emergency situations in real-time, such as the emergency alerts for public safety and situation awareness for personal emergency response. WIPER also employs the SOAP style SOA as its architecture design for integrating the multiple distributed modules with flexible and customizable features that comprise the WIPER system. In addition, WIPER supports multiple approaches for location-awareness such as using cell network and GPS based localization. However, WIPER mainly targets users in governmental organizations who use conventional desktops instead of mobile platforms, and it does not provide any method of collaboration like social networking to mobile users. Thus, it may not be suitable for directly supporting on-site people such as rescuers and victims in disasters.

7. Conclusions and future work

In this paper, we have proposed MS2A, a service-oriented mobile platform for disaster situations, and presented its implementation. By integrating SOA and MSN techniques, as well as the multi-agent framework Aframe, MS2A can simplify and hide the complexity of handing different communication status, varying service requirements of mobile applications for disaster situations. To application developers, MS2A provides a high level software platform to support the efficient and flexible development of diverse and intelligent mobile applications with extensibility for disaster situations. MS2A provides easy and convenient mobile applications with multiple functions to end users, and supports dynamic web service collaboration at run-time, enabling people in disaster situations to easily collaborate with each other through conventional social networking methods, which they are already familiar with. To the best of our knowledge, MS2A is the first solution that fully supports SOA based mobile applications working independently in mobile platforms in an ad-hoc manner. Moreover, a demonstration example has been designed and an evaluation performed to prove the concepts of MS2A system.

There are a number of challenges that need to be addressed in further development of MS2A systems. Although MS2A provides mechanisms to support collaborations of web services at run-time like the location service demonstrated in Section 5, in real disaster situations, the service requirements would be much more complicated with more rapidly changing environments as well. Thus, a systematic mechanism to support dynamic service collaborations for multiple disaster situations needs be investigated. As the SOA based architecture designed for disaster applications are normally based on desktop platforms with central servers that do not work independently as opposed to MS2A on mobile platforms, the conventional evaluation platforms, methodologies for evaluating SOA based systems may not be suitable for the evaluations of MS2A. Therefore, a new methodology for the evaluations of mobile platform based SOA systems like MS2A is needed. In addition, we plan to enable MS2A to work on other mobile operating systems (e.g., iOS and Windows Phone system), and investigate its use in other emergency application areas such as traffic safety and fire alert, etc.

8. Acknowledgments

This work is supported in part by the Canadian Natural Sciences and Engineering Research Council through the DIVA Strategic Network, and by TELUS and other industry partners.

9. References

[1] M. Gaynor, S. Brander, A. Pearce, and K. Post. "Open infrastructure for a nationwide emergency services network", In 5th International Conference on Information Systems for Crisis Response and Management (ISCRAM'08), 2008.

[2] N. Sadeh, J. Hong, L. Cranor, I. Fette, P. Kelley, M. Prabaker, and J. Rao, "Understanding and capturing people's privacy policies in a mobile social networking application", Personal Ubiquitous Computing, August 2009, 13(6) , pp. 401-412.

[3] D. Grace, P. Likitthanasate, "Applications and Business Models for HAP Delivery", Capanina Project Report, 2007.

[4] M. Conti, and M. Kumar, "Opportunities in opportunistic computing", Computer, January 2010, 43(1), pp. 42-50.

[5] Z. Maamar, S. K. Mostefaoui, and H. Yahyaoui, "Toward an agent-based and context-oriented approach for web services composition", IEEE Transactions on Knowledge and Data Engineering, 2005, 17(5), pp. 686-697.

[6] M. Sheshagiri, N. M. Sadeh, and F. Gandon, "Using semantic web services for context-aware mobile applications", Second International Conference on Mobile Systems, Applications, and Services (MobiSys) - Workshop on Context Awareness, 2004.

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

[8] H.W. Lee, and M. Zbinden, "Marrying logistics and technology for effective relief", Forced Migration Review, 2003, 18, pp.34-35.

[9] Y. Jiang, Y. Yuan, K. Huang, and L. Zhao, "Logistics for Large-Scale Disaster Response: Achievements and Challenges", 45th Hawaii International Conference on System Sciences (HICSS), 2012, pp. 1277 – 1285.

[10] D.M. Boyd, and N. B. Ellison, "Social Network Sites: Definition, History, and Scholarship", Journal of Computer-Mediated Communication, October 2007, 13(1), pp. 210-230.

[11] D. Clark, "The design philosophy of the darpa Internet protocols", In SIGCOMM '88: Symposium proceedings on Communications architectures and protocols, 1988.

[12] C. Huang, K. Lan, and C. Tsai, "A Survey of Opportunistic Networks", Advanced Information Networking and Applications Workshops (AINAW), 2008, pp. 1672 – 1677.

[13] S. Ioannidis, A. Chaintreau, and L. Massouli´e, "Optimal and scalable distribution of content updates over a mobile social network", IEEE INFOCOM, 2009.

[14] T. Er, "Service-Oriented Architecture: Concepts, Technology, and Design", Book Service-Oriented Architecture: Concepts, Technology, and Design Prentice Hall PTR Upper Saddle River, 2005.

[15] A. Gupta, A. Kalra, D. Boston, and C. Borcea, "MobiSoC: a middleware for mobile social computing applications", Mobile Networks and Applications, 2008, 14(1), pp. 35-52.

[16] R. Tergujeff, J. Haajanen, J. Leppänen, and S. Toivonen, "Mobile SOA: Service Orientation on Lightweight Mobile Devices", IEEE International Conference on Web Services (ICWS), 2007, pp. 1224 – 1225.

[17] Y. Zhang, L. Zhou, and E. Mao, "A Survey of Mobile Internet Data Management: Models and Searching Methods", Applied Mechanics and Materials Journal, 2011, pp. 738-742.

[18] http://www.brightsideofnews.com/news/2011/2/21/why-nvidiae28099s-tegra-3-is-faster-than-a-core-2-duo-t7200.aspx

[19] http://www.wballiance.com/wba/wp-content/uploads/downloads/2012/07/Ruckus-3G-Offload-Principles.pdf

[20] X. Hu, W. Du, and B. Spencer, "A Multi-Agent Framework for Ambient Systems Development", Procedia CS 5, 2011, pp. 82-89.

[21] X. Hu, J. Zhao, D. Zhou, and V. C.M. Leung, "A semantics-based multi-agent framework for vehicular social network development", In Proceedings of the first ACM international symposium on Design and analysis of intelligent vehicular networks and applications, 2011.

[22] J. Dedecker, T. Van Cutsem, S. Mostinckx, T. D'Hondt and W. De Meuter, "Ambient-oriented Programming in AmbientTalk", In Proceedings of the 20th European Conference on Object-Oriented Programming (ECOOP), 2006, pp. 230-254.

[23] L. Richardson, S. Ruby, RESTful web service, O'Reilly Media, ISBN: 978-0-596-52926-0, Retrieved 18 January 2011.

[24] D. Yazar, and A. Dunkels, "Efficient Application Integration in IP-Based Sensor Networks", Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, 2009, pp. 43-48.

[25] http://pub.kamranzafar.org/files/android/kws/kwspro.pdf

[26] http://paw-android.fun2code.de/

[27] http://code.google.com/p/i-jetty/

[28] http://www.stollmann.de/en/modules/bluetooth-products/bluemod-b20.html

[29] Y. Fang, B. Lee, T. Chou, Y. Lin, and J. Lien, "The implementation of SOA within grid structure for disaster monitoring", Expert Systems with Applications 36, 2009, pp. 5784–5792.

[30] H. Chi, X. Liu, and X. Xu, "A Framework for Earthquake Disaster Mitigation System", China located International Conference on Information Systems for Crisis Response and Management, 2011.

[31] L. Liu, N. Antonopoulos, J. Xu, D. Webster, and K. Wu, "Distributed service integration for disaster monitoring sensor systems", IET Communications, 2010, 5(12), pp. 1777 – 1784.

[32] G. R. Madey, G. Szabo, and A. Barabási, "WIPER: The Integrated Wireless Phone Based Emergency Response System", International Conference on Computational Science (ICCS), 2006, pp. 417-424.

[33] NSF, "Workshop on Dynamic Data Driven Application Systems", 2000.
www.cise.nsf.gov/dddas