SociCare: Towards a Context Aware Mobile Community Emergency System

Mark Bilandzic, Christian Menkens, Julian Sussmann, Daniel Kleine-Albers, Eva Bittner, Armand Golpaygani, Bernhard Mehl, Jonas Huckestein, and Othmane Khelil

Center for Digital Technology and Management
Technische Universitaet Muenchen
80333 Munich, Germany
{bilandzic,menkens,sussmann,daniel.kleine-albers,eva.bittner, armand.golpaygani,bernhard.mehl,jonas.huckestein,othmane.khelil}@cdtm.de

Abstract. Demographic change and the increase in life expectancy continuously increase the average age in our society. Depending on the physical shape and mental constitution, the elderly need assistance to master regular activities in their everyday lives, as well as urgent help in emergency situations (e.g. in case of a collapse or heart attack). Thereby, time is the most crucial factor. In some emergency cases such as heart attack, people might die if there is no immediate help available. This paper presents the design, architecture and prototype implementation of SociCare, a ubiquitous context aware mobile community emergency system. SociCare is designed to that help emergency call centers leverage and coordinate random voluntary helpers nearby the emergency location. It's user interface enables human call center agents to quickly and easily identify, verify and select voluntary emergency helpers based on their context information. Such information can be for example current availability, distance to the emergency location and general skills and ability to provide first aid in the specific emergency case. Based on a prototype implementation of the developed concept an evaluation including for example field and usability studies will be conducted.

Keywords: Ubiquitous Emergency Case System, Mobile Communities, Mobile Web 2.0, Context Awaren, Location Based Services.

1 Introduction

Due to demographic change and increase in life expectancy the average age in our society increases continuously. In Germany for example, people aged 65 years and older will represent around 30% of the population by 2050, which is a 50% increase as of today [6]. This shift will lead to an increasing number of emergency situations and emergency calls in the future [6]. As a consequence, more workforce and money will be required by government and emergency institutions to serve and provide help in emergency situations. A second important aspect

Y. Cai et al. (Eds.): Mobilware 2010, LNICST 48, pp. 338–352, 2010.

regarding emergency services is the first aid response time. For instance, people who experience a heart attack have a much higher chance to survive if first treatment arrives within 90 minutes [7]. Valuable time to the arrival of first aid is lost in detecting and/or reporting the emergency case as well as emergency vehicles losing crucial time due to long access routes and potential traffic jams.

Emergency service providers are in a very challenging position. They need to provide the quickest and best possible emergency service. As mentioned previously, the process of providing first aid is often too slow. In addition, due to the current development of demographic change, health services will face more pressure and costs in serving the increased number of emergency calls. Therefore existing emergency service providers need to be supported and further ways to provide quicker, more immediate and more efficient first aid need to be found.

Our idea to contribute to this development is a service that tracks and gathers context information (e.g. location or availability) of a large community of voluntary helpers and provides it to existing emergency call centers. Such call centers can then use the system to recall and leverage this community's general willingness to provide first aid in emergency situations where they are nearby. Registered voluntary helpers can be selected based on the relation between their situational context and the context of the emergency situation. Thus, the system allows emergency call centers to request their help if their knowledge and current location fits to the context of the emergency situation. If helper's accept the request, they can reach the emergency location within a minimized time frame and provide first aid bridging the time before the ambulance arrives. If the reported emergency case requires only minor or no help at all, they can solve the situation on their own or report a false alarm respectively.

1.1 The Idea: A Ubiquitous Mobile Community Emergency System

The idea of a ubiquitous mobile community emergency system in the previous section bases on two factors: First, the general willingness of people to provide help to others is very high. In Germany for example, according to [8, p. 165] almost one third of the population is committed to some type of volunteer work. Another third of the population is generally interested in providing volunteer work. Their top motives are to help other people and make a useful commitment to the social community [8, p. 176]. We aim to leverage this untapped potential of voluntary helpers that want to support others if they had the opportunity to. Second, in most developed countries mobile phones are widespread throughout the population, and new embedded technology such as high-speed-internet and global positioning technology (GPS) is becoming more and more available for the mobile phone mass market.

Using a combination of such embedded mobile technologies and existing emergency service providers' infrastructures, we aim to close the gap between urgent emergency help seekers and the willingness of nearby potential voluntary help givers. As a first approach to this vision, we have designed the architecture and implemented SociCare, a ubiquitous, context aware mobile community emergency system that manages and provides realtime context information of an arbitrary

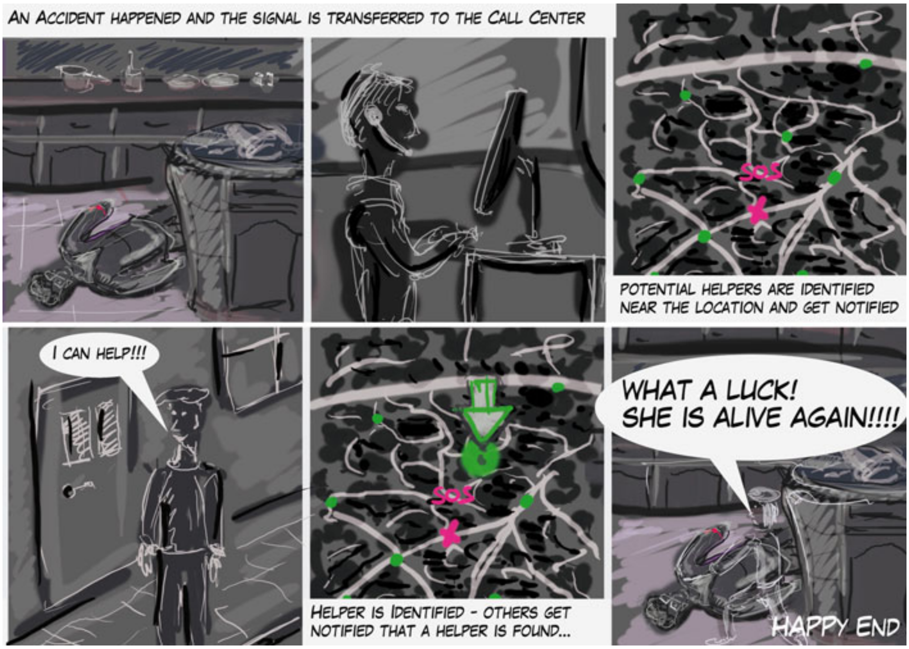

Fig. 1. Use Scenario: SociCare addresses voluntary helpers to provide first aid in emergency cases before the ambulance arrives (own illustration)

number of certified voluntary community emergency helpers. The context information managed by SociCare includes location and distance to an emergency situation as well as the availability and ability (i.e. skills) of a voluntary helper to provide first aid.

By doing so, SociCare enables emergency call centers to quickly and easily identify, verify and select voluntary emergency helpers that are able to provide aid in an emergency situation. As a result, SociCare contributes to the above mentioned problem in the following ways:

- Response time: By the integration of voluntary helpers and random passers-by in emergency situations, SociCare enables emergency services to reduce the response time from receiving the emergency call to having someone to help at the location of emergency. Thereby, the emergency call can be triggered by the help seeker himself, or an emergency detection sensor attached to the person's body.
- False alarms: It helps emergency services to reduce costs by integrating a cost free community of voluntary helpers that can detect costly false alarms early at the point of interest, thus avoid e.g. costly emergency vehicles to be sent.

– Coordination: The system integrates a Web 2.0 enabled interactive map interface that enables emergency call centers to easily identify, communicate with and coordinate voluntary helpers in real time (see Figure 1). With its adaptable database infrastructure and web based interface, this solution can be easily built upon the existing infrastructure of widespread existing call centers.

2 Related Work

Providing help to elderly people and serving emergency situations has been targeted by previous work. These proposed solutions focus on emergency detection, such as industrial products that provide body sensors measuring and detecting anomalous vital signs or other emergency situations (e.g. a sudden collapse). In a similar context, previous research studies have focused on integrating various sensor types and emergency detectors in an extensible middleware architecture [4] [5] [3] towards a solution for extensive in-house safety. However, such approaches do not cover the issue of late first aid arrival in emergency cases. Up to our knowledge, the only previous work that is based on the principle of integrating a community of voluntary helpers in the emergency process is presented in [3]. The proposed system AGAPE proposes a similar concept for realizing a ubiquitous mobile community emergency system to SociCare. Same as SociCare, AGAPE leverages mobile clients as well as location- and proximity detection technology in order to identify potential nearby helpers. However, AGAPE focuses on a purely automatic group formation, coordination and management of outdoor voluntary helpers. In contrary, we believe that this complex task can be better performed by a human operator. Therefore, with SociCare we put our focus on providing an efficient tool and user interaction to enable a human emergency call center agent to quickly identify, select and notify the best suitable and available voluntary emergency helper depending on the context of himself and the emergency situation.

3 SociCare Design

The design of the SociCare system is twofold. It covers (1) a web front end to be used by a human agent at the emergency call center and (2) mobile clients that run as mobile applications on the voluntary helpers' devices. Therefore, the overall architecture is based on a client-server architecture. The SociCare server communicates with the call centers' servers and the voluntary helpers' mobile clients. This way the helpers can be sure that their position is only visible to the call centers in case of an emergency.

As the application is time-critical by use case, the emergency notifications should be transmitted quickly and reliably to the helpers. This suggests some kind of push mechanism. The call center agent should be able to immediately view the location of potential helpers around an emergency. This means that the SociCare server has to be aware of all helpers' locations at any time.

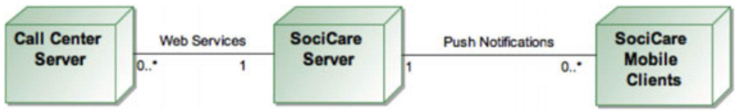

Fig. 2. SociCare connects existing emergency call centers and the community of voluntary helpers (own illustration)

3.1 Web 2.0 Map Client for the Emergency Call Center

In emergency call centers advanced IT support is already widely established. Operators have e.g. the possibility to detect the position of an emergency quickly by the information they receive from the caller or emergency sensor, thus they can send an emergency vehicle closest to the place of interest. Therefore we decided not to provide a stand-alone emergency platform, but rather integrate the ubiquitous mobile community of voluntary helpers as an add-on to the existing services. Integration within the existing systems is supposed to save costs and increase the likelihood of adoption. The value added by SociCare to emergency call centers lies in the database of registered voluntary helpers that can be accessed from the standard platform interface. Features include a real time map-display of the emergency location and nearby voluntary helpers. As helpers are registered in the SociCare database, additional information on e.g. their profession, their skills and abilities to provide first help or their mobility can be shown, if available to allow the choice of the best helper suitable for the respective emergency case. Call center agents can view helper's details, select the closest by location and inform them by push notification on their mobile phones. Visualization and usability is critical. Therefore the map on the agent's side shows in clear symbols where tracked people are and if they agreed to help or declined a query. In case one helper agrees to take the task, one more key feature is the communication between the call center and the helper. The agent is then able to call the helper directly from the platform to guide him to the emergency location and give instructions how to provide first aid.

3.2 Mobile Phone Application for Voluntary Helpers

Helpers who register for SociCare provide their contact information and willingness to help in emergency situations. Thus, it is crucial for them that the sign up process is simple and user friendly and their data is being held by a trustworthy organization. Therefore we intent the SociCare system to be run by the call centers themselves or a trusted organization that acts as a mediator between the voluntary helpers community and a symposium of different call centers. In the latter case, one would also benefit from network effects and scalability of the service.

The mobile client is based on the Google Android mobile phone platform. Voluntary helpers can join the network by simply downloading the SociCare application to their phones and providing some additional information about their profession and skills in providing first aid. In case an emergency in their surrounding happens, the call center agents can track their current location and will inform them conveniently by push notification. If the helper feels able to provide help, he can agree to accept the request. As helpers might want to help but fear to do something wrong in the help giving process, they can request a voice call and be guided by the call center agent while providing first aid. Furthermore, data security is crucial; so all helper's context information, location data and emergency specific details have to be kept save and protected from misuse.

4 SociCare System Architecture

SociCare is a distributed system with two different types of clients, different communication channels and protocols for each type and a central application server. This section illustrates all architecture decisions and ideas for the SociCare deployment and communication architecture as well as the server and client side architectures.

4.1 Deployment and Communication Architecture

The general deployment and communication architecture of the system is based on a standard Client-Server architecture concept which defines one central server that manages one or many secure communication channels with an arbitrary number of different types of clients. Figure 3 shows an overview about the SociCare Client-Server architecture.

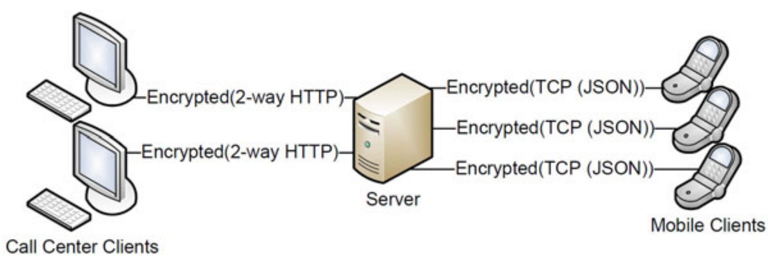

Fig. 3. SociCare Client-Server Architecture and Communication Channels (own illustration)

The server hosts the SociCare application logic, all SociCare related data and handles all data communication with all clients using two different types of SSL (or similar) encrypted communication channels:

– **Call Center Channel** - Represents a steady data connection between the SociCare server and multiple clients on multiple call center workstations. The channel uses the Hyper Text Transfer Protocol (HTTP) protocol [9] to enable call center clients to render the web browser based GUI of the

SociCare application. Since HTTP is a protocol that allows clients to request information from servers but does not allow servers to push information to clients, a HTTP-Push [10] concept is used to enable a two way data communication between the SociCare server and the web browser based call center client.

– **Mobile Channel -** Represents a steady data connection between SociCare server and multiple mobile clients of voluntary helpers. Transmission Control Protocol (TCP) is used as two way data communication protocol and the JavaScript Object Notation (JSON) [11] is used as payload data format. This format was chosen for two reasons: i) it is lightweight and therefore has a comparably low data volume, which might be important for the users on expensive mobile contracts, ii) it is easy to encode and decode from any object.

The communication architecture of SociCare defines a stable and extendable application logic process and communication protocol between all types of clients and the central server. Figure 4 illustrates a simplified standard SociCare emergency process and communication sequence without error message and component failure handling.

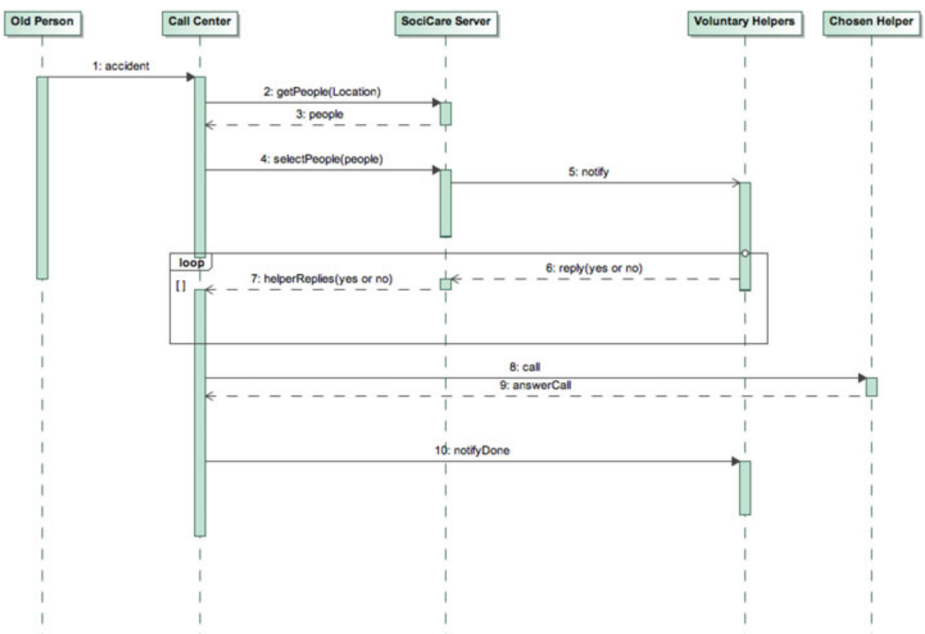

Fig. 4. SociCare Communication Architecture (own illustration)

In this sequence example an "Old Person" indicates an emergency situation (1) which the emergency call center receives. The call center uses the SociCare call center client application to trigger a request (2) to find all voluntary helpers in the area of the emergency. The server responds (3) with a list of voluntary helpers including data about their certification details and ability to help immediately. The call center agent selects the most appropriate voluntary helpers from that list (4) and notifies (5) them on their mobile devices. The SociCare server collects (6) the responses (yes - available; no - not available) of all notified voluntary helpers and forwards (7) them to the call center client immediately. The call center agent selects one or a group of available helpers and informs (8) them about the emergency situation details. These helpers rush to the emergency situation location and help the person in need. After the situation was solved, they inform (9) the call center. The call center notifies (10) all other voluntary helpers that the situation was solved.

4.2 Server Side Architecture

The server side architecture of the SociCare application is a Model 2 web application architecture that defines a Model View Controller (MVC) [1] concept for web based applications. By applying the MVC architecture to a web and mobile based application, data model details are separated from the presentation and the application logic and processes that use this model. Such separation allows multiple views to share the same model, which makes supporting multiple clients easier to implement, test, and maintain [1]. This is essential for the SociCare system since the different types of clients have to work on the same data model and the overall SociCare system has be extendable with new services, applications or different types of clients in the future.

Figure 5 illustrates the dependencies between the SociCare MVC components:

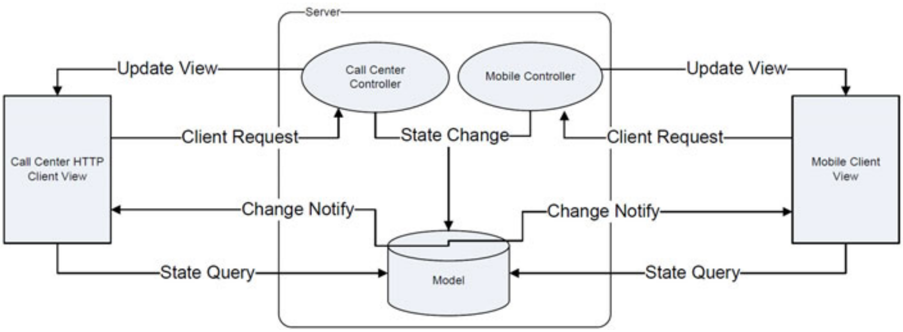

Fig. 5. SociCare Server Side MVC Architecture (own illustration)

– **Model** - The model represents SociCare application specific data and rules that govern access to and updates of this data. Often the model serves as a software approximation to a real-world process, so simple real-world modeling techniques apply when defining the model. The SociCare model stores data and access rules for SociCare call center user accounts, all registered voluntary helper accounts including details about their mobile devices. Further

more, it manages a real time data set of emergency situations, their current status, communication details and up-to-date context data of all voluntary helper mobile clients that are connected to SociCare. In addition, the model stores all logging and event history information of the SociCare system, so every action, event, interaction and task is logged and can be tracked.
- **Views** - SociCare defines two different types of views, workstation based call center client views and mobile client views. The views access the data within the model, define and specify how the data needs to be presented and render the data on the clients. The views are responsible for maintaining consistency in their presentation when the model changes. This can be achieved by using a push model, where the view registers itself at the model for change notifications to query the model. Or a pull model, where the view is responsible for querying the model regularly to present the most current data. Due to the fact that SociCare works with real time context data of mobile clients and emergency situations, a pull model is not efficient enough and only a push model is able to update the views right at the moment when updated data is available. In addition to rendering model data, the views present the GUI of the application and initiates user interaction requests (including entered user data) to the controller of the application.
- **Controller** - The controller translates interactions with the GUI on the view into actions to be performed on the model. The actions performed on the model include activating application logic processes or changing the state of the model. Based on the user interactions and the outcome of the actions, the controller responds by updating the current or changing to an appropriate view. Thus, the controller centralizes functions such as view selection, security, and templating, and applies them consistently across all views. Consequently, a major advantage of this architecture is, when the behavior of these actions needs to change, only a small part of the application needs to be changed: the controller and its helper components [2].

  SociCare defines two completely different types of clients that require different corresponding actions, processes, permissions and configurations. So two controllers, one for call center clients and one for mobile clients have to be defined. Each of these controllers only implements the application logic and processes that are needed to fulfill all tasks for the client it serves.

4.3 Client Side Architecture

The mobile client represents the user interface for voluntary helpers and needs to have a stable two way data connection to the SociCare server as well as access to up-to-date context information such as the current geographic location. Figure 6 shows an overview of the mobile client architecture.

The mobile client application is a native application on the mobile device and consist of four different components:

- **Application Logic and GUI** - Implements all application logic code and the mobile platform specific GUI for the voluntary helper application.

- **Data Store** - Implements a central data storage component on the mobile client for all configuration and application data such as user account settings, location information history, etc.
- **Location Service** - Implements a parallel running process that accesses the mobile devices operating system (OS) in order to retrieve up-to-date geographical location information from the location hardware built into the device.
- **Connection Service** - Implements a parallel running process that manages a stable two way connection between mobile client and server. It takes care of session negotiation, sending and receiving messages as well as dealing with connection losses and reconnecting etc.

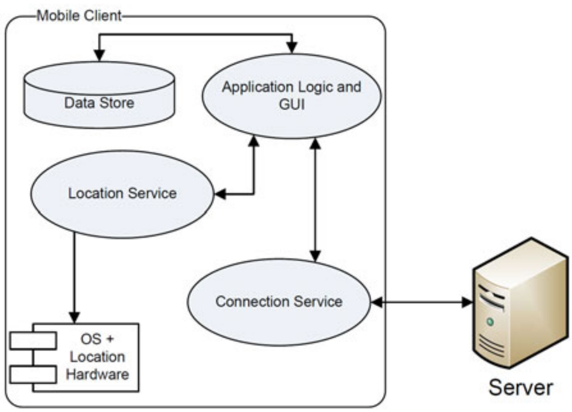

Fig. 6. SociCare Mobile Client Side Architecture (own illustration)

5 SociCare Prototype Implementation

In order to show the advantages of the architecture decisions above and to being able to use a running system in future studies and evaluations, a proof of concept prototype implementation of the SociCare system was implemented. The following section outlines some details about implementation and development platform decisions and illustrates the client GUIs.

5.1 Server

As server platform Apache Tomcat 6.x [12] was selected and both controllers as well as the model were implemented as Java J2EE [14] application components. The mobile client TCP communication channel was implemented in a standard Java component using third party JSON encoding and decoding libraries. Since the call center communication channel requires a non standard HTTP-Push concept to function properly, several options were considered and the open source technology BlazeDS was used.

BlazeDS is a server-based Java remoting and web messaging technology that allows an application to connect to the back-end and push data in real-time to Adobe Flex and Adobe AIR Rich Internet applications (RIA). The Message Service provides a complete publish/subscribe infrastructure allowing Flex clients and the server to exchange messages in real time. Remoting allows a Flex application to directly invoke methods of Java objects deployed in an application server. [13]

5.2 Call Center Client

The SociCare call center client was implemented using the Adobe Flash based Adobe Flex technology which perfectly integrates with the HTTP-Push server technology BlazeDS. Flex enables the implementation of stateful rich web browser based applications where significant changes to the view or sending and loading data to and from a server do not require reloading the current view. Flex comes with a set of user interface components including buttons, list boxes, trees, data grids, several text controls, and various layout containers. Further more, advanced features like web services, drag and drop, modal dialogs, animation effects, application states, form validation, and other interactions go far beyond HTML web application possibilities and make Flex the perfect choice for the SociCare call center client prototype implementation. [15] Figure 7 shows the main screen of the Flex based SociCare call center client.

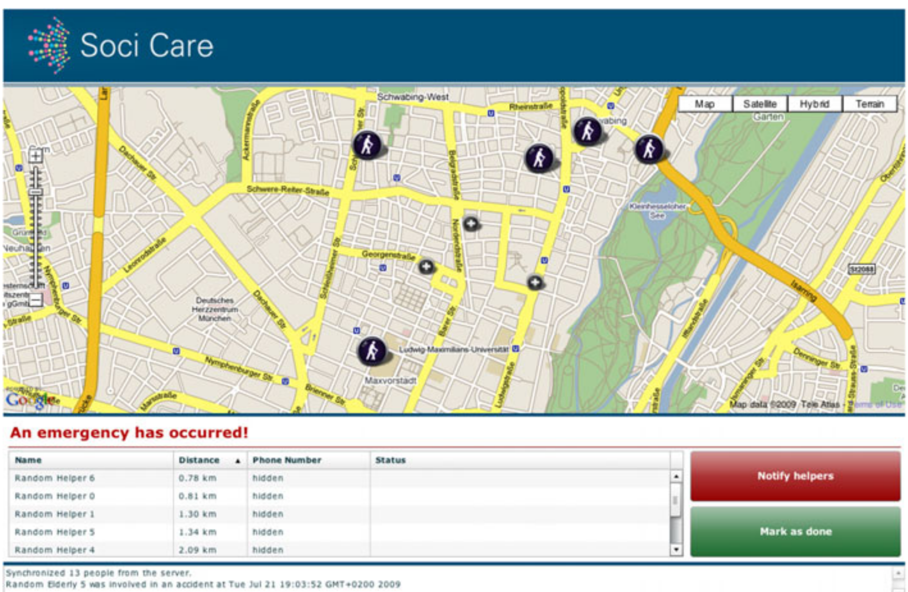

Fig. 7. SociCare Flex based call center client (own illustration)

The call center client uses a map with up-to-date mobile client location information as main user interface component. The agent is able to select helpers and view detailed information about them as well as to notify them about an emergency situation. By interacting with a mobile user's symbol on the map, the agent is able to communicate with the user and so coordinate the group of helpers. Finally, the agent is also able to mark an emergency situation as completed which informs all helpers on their mobile devices.

5.3 Mobile Client

The SociCare mobile client for voluntary helpers was implemented using mobile devices based on the Google Android platform [16]. Android provides a mature support for accessing the geographic location information of the mobile device, for parallel processes for the location service and connection service as well as for two way TCP network connections between client and server. Encoding and decoding of the JSON payload of the messages was implemented using third party libraries.

Figure 8 illustrates two screens of the SociCare mobile client. Each screen renders a view and handles all interactions with the user.

Fig. 8. SociCare Mobile Client - Position screen and Settings Screen (own illustration)

The Position screen shows the own most up-to-date geographical location on a map. The Settings screen renders an input form for modifying the mobile client communication channel settings Host Internet Protocol (IP) address, Host-Port, Username and Password.

Figure 9 illustrates the two most important screens that are rendered when an emergency situation occurred and the mobile client is notified. The Emergency screen renders an overview map that shows the own most up-to-date geographical as well as the geographical location of the emergency. Also, the screen includes a dialog that allows the helper to state his availability to help.

If the helper is not available and selects "No", the screen disappears and the Position screen is rendered. If the helper is available and selects "Yes", a Emergency Routing Dialog is rendered on top of the map and offers to connect to the Google Maps Routing service in order to route the shortest and/or fastest path to the emergency.

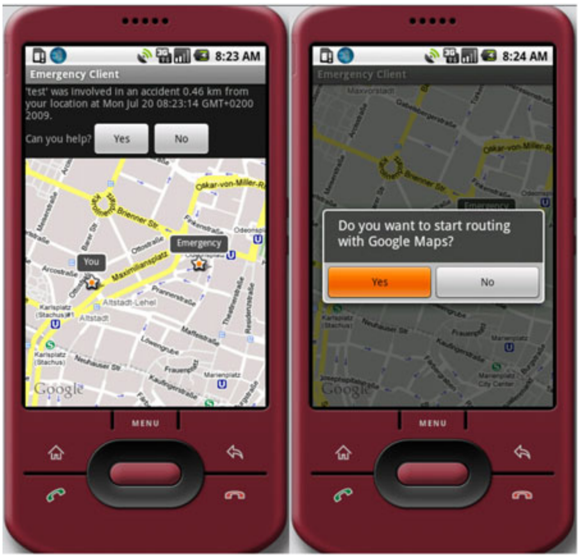

Fig. 9. SociCare Mobile Client - Emergency Screen, Emergency Routing Dialog (own illustration

6 Conclusion and Future Work

This work aims to close the gap between urgent emergency help seekers and the willingness of nearby potential voluntary help givers. From this background, we have designed the architecture and implemented SociCare, a mobile community emergency system that tracks real-time context information of certified voluntary helpers, and visualizes this information on an interactive map interface for emergency call centers. After receiving an emergency call, the human agent in the call center can identify and request nearby voluntary helpers who are equipped with the SociCare mobile client to help the person in need. The context information managed by SociCare includes the distance to an emergency situation as well as relevant context information, such as availability and skills of the voluntary helpers to provide first aid. We have designed SociCare that it can be easily integrated with the existing emergency management software used in the call centers. As to our knowledge this approach of a ubiquitous mobile community emergency system attached to existing emergency call centers is the first of its kind, we plan to undertake further research on how well it performs in real use cases. Therefore, we aim to pursue usability studies on both provided interfaces: (1) the interactive map interface needs to be tested with real call center employees and evaluated how quick and efficient they perform in selecting and interacting with the identified voluntary helpers. (2) On the other side the mobile client's interface needs to be evaluated on usability factors in real context. We believe that such a user study under real emergency conditions, i.e. the user experiencing a stress situation, is crucial, since their performances will differ from usual labor setting tests that we have performed throughout the development phase.

Acknowledgment

The authors would like to thank Deutsche Telekom Laboratories for providing the prototyping equipment for SociCare as well as their technical support and feedback. Furthermore, we thank the anonymous reviewers for their feedback and input on earlier versions of this paper.

References

1. Singh, I., Stearns, B., Johnson, M.: Designing Enterprise Applications with the J2EE Platform. Sun Microsystems, Inc. (2003)
2. Bien, A.: J2EE Patterns. Addison Wesley, Reading (2003)
3. Bottazzi, D., Corradi, A., et al.: Context-Aware Middleware Solutions for Anytime and Anywhere Emergency Assistance to Elderly People. IEEE Communications Magazine 44(4), 82–90 (2006)
4. Pung, H.K., Gu, T., et al.: Context-Aware Middleware for Pervasive Elderly Homecare. IEEE Journal on Selected Areas in Communications 27(4) (2009)
5. Taleb, T., Bottazzi, D., et al.: ANGELAH: A Framework for Assisting Elders at Home. IEEE Journal on Selected Areas in Communications 27(4), 480–494 (2009)
6. Statistisches Bundesamt Wiesbaden, Bevoelkerung Deutschlands bis 2050. Ergebnisse der 10. koordinierten Bevölkerungsvorausberechnung (2003)
7. Wire, H.: Philips seeks to reduce time from heart attack to treatment (2007), `http://www.healthtechwire.com/Pressrelease.146+M5a1042be3f1.0.html` (accessed 01.02.2010)
8. BMFSFJ, Altenhilfestrukturen der Zukunft - Abschlussbericht der wissenschaftlichen Begleitforschung zum Bundesmodellprogramm, Bundesministerium für Familie, Senioren, Frauen und Jugend (2004)
9. Fielding R.: Hypertext Transfer Protocol - HTTP/1.1 (1999),`http://www.w3.org/Protocols/rfc2616/rfc2616.html` (accessed 01.02.2010)
10. Google, ServerPush (2007), `http://code.google.com/p/google-web-toolkit-incubator/wiki/ServerPushFAQ` (accessed 01.02.2010)

11. Crockford D.: The application/json Media Type for JavaScript Object Notation (JSON) (2006), http://www.ietf.org/rfc/rfc4627.txt (accessed 01.02.2010).
12. The Apache Software Foundation, Apache Tomcat (2009), http://tomcat.apache.org/ (accessed 01.02.2010)
13. Adobe Open Source, BlazeDS (2009), http://opensource.adobe.com/wiki/display/blazeds/(accessed 01.02.2010)
14. Sun Microsystems, Java 2 Platform - Enterprise edn.(2009), http://java.sun.com/j2ee/overview.html (accessed 01.02.2010)
15. Adobe Labs, Adobe Flex Framework Technologies (2009), http://labs.adobe.com/technologies/flex/ (accessed 01.02.2010)
16. Google, Android (2009), http://developer.android.com/index.html (accessed 01.02.2010)