# Mobile Location Based Services for Trusted Information in Disaster Management[1]

Lemonia Ragia, Michel Deriaz and Jean-Marc Seigneur

**Abstract**. The goal of the present paper is to provide location based services for disaster management. The application involves services related to the safety of the people due to an unexpected event. The current prototype is implemented for a specific issue of disaster management which is road traffic control. The users can ask requests on cell phones or via Internet to the system and get an answer in a display or in textual form. The data are in a central database and every user can input data via virtual tags. The system is based on spatial messages which can be sent from any user to any other in a certain distance. In this way all the users and not a separate source provide the necessary information for a dangerous situation. To avoid any contamination problems we use trust security to check the input to the system and a trust engine model to provide information with a considerable reliability.

## 1 Introduction

The wireless technology becomes important in our daily life because it provides a lot of services. The World Wide Web gives the opportunity to people to connect mobile phones or portable devises to Internet. Universal Mobile Telecommunication System (UMTS) with the new smart phones enable more services. The number of people that use the Web and the wireless technology is increasing rapidly.

User location was difficult to find out but with the usage of Global Positioning System (GPS) new possibilities are open. The integrated technology of GPS devices gives the location of the people quickly and with accuracy. That means that we can have location based services (LBS) which connect, in principle, the geographic location with user requests.

There are several approaches that show personalized LBS services for different applications: in the area of tourism [14], [1], [16] or navigation [10]. There are also some approaches for LBS which discuss the connection to databases [7], [6].

Disaster management is an important topic for local authorities, governments and disaster managers because they try to manage efficiently all the information provided mainly from people on the field to provide directions to the public. LBS for disaster management is extremely useful for the citizens since they can have great benefits having the right information in the appropriate time. In the scientific area of disaster management there are approaches which simulate a pre-disaster phase [9] or demonstrate an open source software especially for natural hazards [4]. Application for LBS for disaster management can be found on the area of

[1] A revised version of this work is published in the Proceedings of the IEEE International Conference on Information Systems Development

health care [13]. There are different aspects for services for disaster management. We can classify them in the following categories:

- Services for Natural hazards

This service provides information about the natural physical phenomena which can happen any time. Earthquake, flood, cyclones, fire etc., belong to this category. These information use *historical data* and try to make prediction for local authorities or other responsible offices to share the information and advise people how to avoid such a situation and protect themselves.

- Safety related services

In this category the information is related to the safety of the people in unexpected events. Man-made disasters such as car accidents or a plane crash are included. It provides information for a dangerous situation and it uses *real time data*. These *real time data* are related to this event and can be provided by any user.

In our approach we deal with the safety related services. An important issue in disaster management is for instance the traffic control. This service gives information about a safe and free travel and helps the users to avoid any kind of unexpected difficult occasion. It does not include the normal traffic jams during rush hours but it is related to unexpected events happening in special conditions. It takes into account a big area of infrastructure and it is updated by the users living through the event.

The mobile LBS application in our system is based on *spatial messages*. A *spatial message* is a message which refers to a specific geographic location. It allows a mobile user to publish a geo-referenced note so that any other user close and affected can get the message. Let us consider a community of car drivers. For example, an accident can happen or there is a fire next to the road. The car drivers would like to communicate about such event related dangers in specific places.

*Spatial messaging* has been already used. We could site for instance E-Graffiti [2]. E-Graffiti is a spatial messaging application that allows a user to read and post geo-localized notes. These notes can be either public or private, meaning that only the set of people defined by the author are able to read the note. E-Graffiti has been designed to study the social impact on spatial messaging.

Another interesting example is GeoNotes [12]. GeoNotes has more functionalities than E-Graffiti. While posting a note, the user can choose how he is going to sign it (for privacy reasons the user can write any text he wants as a signature), decide whether people are allowed to comment on it, and decide whether anyone can remove this message. For the readers, the graphical interface of the application provides some interesting functionalities like showing all the neighbouring messages or sort them according to different criteria. Inspired by the E-Graffiti evaluation, GeoNotes discarded the remote authoring of tags as well as the possibility to "direct" notes to certain users.

In our system the mobile location based services include the connection to a central database and in principle every user can send data to the central database using *virtual tags*. The *virtual tags* include any spatial messages which are related to a Geo referenced context related to disaster information. An important issue in our system is the use of a trust engine which gives information with considerable reliability to the users. We develop a framework that provides, among other things, a set of generic trust engines and a tool box providing geo-related tools.

This framework, called LBSDisMan (Location Based Services for Disaster Management), should provide APIs (Application Programming Interfaces) in order to ease future development of applications using *virtual tags*. The results can be presented in a cell phone or any other internet appliance.

In the next section we present the system architecture and give some details for the server and the client part. Section 3 gives an example of a real application and section 4 discusses results and further work to improve the system.

## 2 System Architecture

In order to share the spatial data among the users, we use a centralized architecture. The data is organized in small units that we call *virtual tags*. Each tag contains geo-related information, that means its position, and a content that is written in HTML.

The server part of our framework is represented in figure 1. The application designer starts by choosing the trust engine according to the kind of tag he is dealing with, then customize it with code (if needed) and parameters, and finally defines how the tags have to be stored (memory, flat files, database). For the storage, template classes should be provided in order to ease the development but still let the possibility for the developer to implement his own specifications.

The trust engines are generic and easily extensible. Each trust engine proposes a set of parameters in order to adapt its behaviour according to a given application, and all the trust computations are made in a standard and formalized way. This means that an application designer is able to adapt a trust engine by adding, modifying or removing the rules used to compute a trust value. Roughly speaking, the designer of a new application will have to code "how much a specific behaviour in a specific context costs in terms of trust value". He will therefore only have to code behaviours directly related to its application, leaving the framework doing all the job of maintaining and managing the trust information. This should guaranty that our trust engines can be adapted to any situations, and therefore really be generic.

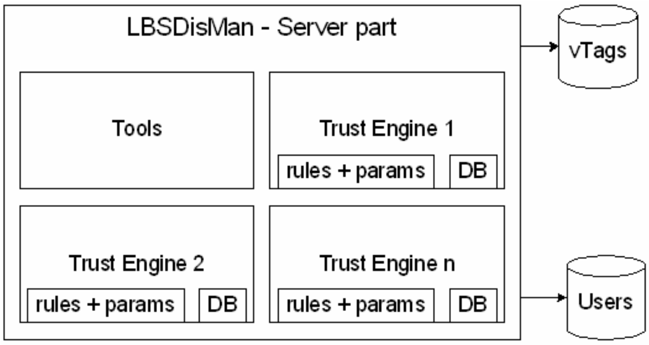

***Figure 1*:** LBSDisMan server part of the framework

The Tools box is used by the trust engines and can also be accessed by the application. It contains mostly geographical related tools, like methods allowing conversions or methods handling tags of different formats.

All accesses to the database (vTags contains the *virtual tags* and Users contains the ID of the users) are done via the trust engines. It can of be any storage solution, including no permanent storage (information is kept in memory), flat files or a SQL standard database. Throughout this document, we will use the term "database" or its abbreviation "DB" to mention any storage system, and use the term "SQL database" or "SQL DB" if we talk about a "traditional" relational database using the SQL language to interact with.

Each trust engine provides a box allowing personalization it through rules and parameters, as well as a DB box responsible to store the tags in a permanent way. The latter should provide classes that can be adapted for the main storage architectures but also provide a generic solution that can be extended by a developer willing to implement its own storage architecture.

The trust engine should be accessed via three main primitives:

**setTag**. This primitive simply creates a new tag. No trust mechanism is used.

- **getTags**. Returns a list of tags. The requester specifies which filter he wants to apply to the result. For instance, a user can ask to get all the tags in a certain radius, with updated trust values for the author and the reviewers, and let the application decide what to do. But he can also ask to get only the tags that are above a certain trust level and ignore the others. Or he can apply a personal filter and not use the trust mechanism at all, like asking all the tags that are authored or reviewed by a user.

**reviewTag**. Reviewing a tag means to rate it, optionally to add a comment, and then update the trust tables of the reviewer, the author and the former reviewers. The way the trust tables are updated is defined through the rules and the parameters. The framework splits all the behaviours so that the application developer can simply write the rules according to the needs of its application.

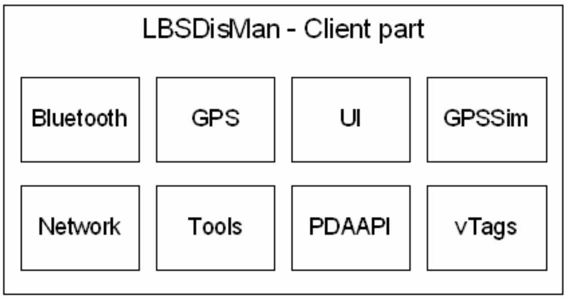

***Figure 2:*** LBSDisMan client part of the framework

The LBSDisMan framework provides also an API for the client part. This API provides geo-related tools, tools to manage virtual tags, and also some general tools that will be needed by spatial messaging applications like sending information over the Internet from a mobile device, storing information on the local device, or accessing to an external or internal GPS (or another positioning device). A graphical representation of the client part is given in figure 2.

### 2.1 Security in the System

In a secured spatial messaging system, a user can be sure that the message he is reading is really written by the mentioned author, that nobody has modified the content of the original message, and that all other available messages at this place are available. More precisely, a secured spatial messaging system has to respect the "traditional" security services that are [3]:

- Confidentiality. Protection of the information against divulgations.
- Integrity: Protection of the information against modifications.
- Availability: Information is always available.
- Entity authentication: The author can be identified.
- Data origin authentication: Information can be linked to its author.
- Non-repudiation: The author cannot repudiate a message.
- Non-duplication: Protection against copying the information.
- Anonymity: The real-life identity of the users must be preserved.

Our aim is to focus on specific security services, the ones that are required for spatial messaging (in addition to the "traditional" ones). These are centered on the *pseudonym concept* [8]. What we would like is a system in which an author can be identified, but at the same time we would like to prevent any link with his real-life identity. A new user is therefore able to get a pseudonym in an anonymous way, but only one. If the person can obtain an unlimited number of pseudonyms, then the system can be victim of a Sybil attack [5]. The user must also be able to change its pseudonym. Again, this must be done in an anonymous manner and it must be impossible to link a former pseudonym with the new one.

A secured spatial massaging system must therefore respect, in addition to the "traditional" security services, the following "specific" ones:

- A user has only one pseudonym at a time.
- A user must be able to change its pseudonym.
- It is impossible to link a pseudonym to a real-life identity.
- It is impossible to link two pseudonyms of the same real-life identity (an old one with a new one).

Each pseudonym is unique, it is impossible that two different real-life identities share the same pseudonym. This is even true over time; if a user changes its pseudonym, the old one is locked and can never be used again. If in our application, we have a small community of users, we could choose to base the security of the database, its access and the users information via the use of a Public Key Infrastructure (PKI).

### 2.2 Trust in the System

The previous section discussed the security aspects of spatial messaging. A reader can be sure that a given message is really posted by its signer and that the content has not been modified since. But even if the reader can be sure about the author's identity, it is useless if they do not know each other. This section discusses how to add trust information on spatial messages so that the reader can evaluate the reliability of a message.

Trust is a very complex concept. Even if it is part of everyday life, different people give also different definitions of what trust is. This observation is even strongly accentuated when we try to explain how to build a trust relation between machines, or between humans and machines. One reason is that most models are only designed and specialized for peer-to-peer files sharing systems. For example, these models do not take time into account. In spatial messaging time is very important. For example a message indicating a high risk of avalanches posted yesterday has to be taken more seriously than the same message posted six months ago.

Spatial messaging needs a specific trust model that takes time into account, as discussed previously, and that is sufficiently flexible to be adapted to different situations. For example, in a mountain guide example, we suppose that the community of users is quite small and that a Web-Of-Trust trust model [14] will be sufficient. If user A trusts user B at 0.8 (out of 1), and user B trusts user C at 0.5, then user C rating (in user A 's eyes) will only count for 0.8 * 0.5 = 0.4. This does not mean that user A 's trust in user C is only 0.4. It is only the number by which user C 's rating will be multiplied.

However this model does not work for large communities. In this case we need to know the global reputation of the author. We could of course provide two different models depending on the size of the community. There is also a third trust model, the one that informs about the reliability of the message itself, without taking care of the author's reputation. Even a very reputable author can make a mistake and publish wrong information. Or, even more likely, a message signed by a reputable editor can contain outdated information.

We use a trust model which is actually the model that will combine the former ones. Its role is to answer the "How to trust the different trust models" question. The three previous models will give us three different trust values, and the fourth model's role is to determine, according the current situation, how much weight to give to each value. In this way we obtain a trust engine that is generic and can be easily applied to any situation.

## 3 Implemented Prototype

We have developed a system for Mobile Location Based Services for Disaster Management according to the architecture outlined in the previous section. We applied it in a specific topic of disaster management which is road traffic control. We used a central database including traffic data and additional data related to unexpected events and disaster phenomena. For the geometry we follow the standards of Open Geospatial Consortium [11] using their geometrical attributes.

The system incorporates spatial queries including requests regarding the content, the geographical position, the address and the time. The content describes all the information about this specific theme, e.g. “give all the fires in a distance of 50 km of the place with coordinates X and Y”. The geographical position is based on the longitude and latitude of a location which can be taken by a navigation system or

GPS, e.g. a spatial query can be “car stops at a position with X and Y coordinates, is there any problem in the highway”. An address refers to a street (name, number or code), to postal code, to a name of a city, e.g. “give all information about traffic jam in the highway number 5 in Switzerland” and the date can be a day, a month, week, year, part of day, hour, minute, e.g. “show all the accidents positions during April 2006”. Or when car driver is on a forest road can ask “is any fire in a specific part of a road”. The content will be also chosen by the service and there will be a lot of possibilities to make different kind of queries.

The system allows the user to enter data via virtual tags using a location. Then spatial messages can be sent to other users. In this case a user can ask for a user profile. The trust engine provides a trust value which is between [0,1]. The zero means totally unreliable, 0.5 neutral and one highest reliability. We developed an application running on mobile phones that helps the user to find the closest exit from the centre of a city (fig. 3). Something unexpected happened and the traffic stops for some time. A Bluetooth GPS connected to the mobile phone gives the current position of the user, and the GPRS protocol is used by the mobile in order to connect to the server that hosts the data. The user receives spatial messages in his cell phone “there are flames in a building” with high trust value.

In this example the user using a GPS system provides his/her coordinates to the system and ask the query “which are the next exits from this specific location in the centre of Geneva”. The green point with a circle shows the position of the user. Then the system shows the exit 1, 2, 3 with blue colour (fig. 3). Due to the security part of the system and after analysing the data the system gives only one solution to the user which is the number 3 in this case and highlighted in green on the mobile phone display. The visualization of the results can be displayed in a mobile electronic device like a cell phone (fig. 3).

The implementation of our system includes spatial queries in SOAP protocol and in XML language and answers can be shown via XML or SOAP. The system uses suitable methods for selecting, storing and detecting user profiles according to their location.

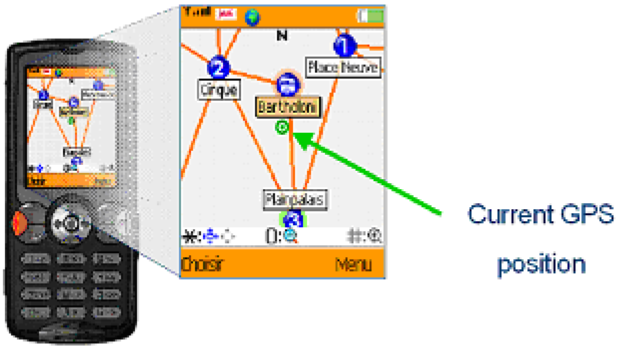

***Figure 3:*** Visualization of the information

## 4 Conclusions and Future

We present a system for mobile location based services for disaster management and its application for traffic control. The system uses spatial messages to share geo referenced information to the users. It incorporates a central database and every user is allowed to feed data in the database. The users can use the services to ask queries at a given spatial location and receive the messages real time in a smart phone or other Internet device. Our system integrates trust engines and its security is taken also into account. In this way we improve the quality and reliability of the services. We implemented a disaster management scenario using real examples and we used the cell phone display to show the results of the spatial messages.

Currently we work on the implementation of the designed services to improve the results. We would like to use a bigger scenario with more real data. We envision a system applied in other applications of disaster management. From the database perspective we will work more in database integration and try to use the system with real data provided by other sources. In addition, we investigate the model of the trust engine as a general framework for open applications.

## References

[1] Antikainen, H., Rusanen, J., Vartiainen, S., Myllyaho, M., Karvonen, J., Oivo, M., Similä, J. & Laine, K., 2006: Location-based Services as a Tool for Developing Tourism in Marginal Regions. Nordia Geographical Publications, 35: 2, pp. 39-50.

[2] Burrell, Jenna, Gay, Geri K. 2002: E-graffiti: evaluating real-world use of a context-aware system. In Interacting with Computers, 14 (4) pp. 301-312.

[3] [Charton E., 2005: *Hacker's Guide, Edition DeLuxe*. Campus Press.

[4] Currion P., Silva de C., and Walle Van De B., 2007: Open Source Software for Disaster Management. *Communications of the ACM*, Vol. 50, Issue 3, pp. 61-65.

[5] Douceur J.R., 2002: The sybil attack. In Proc. of the IPTPS02 Workshop, Cambridge, MA USA, March.

[6] Gruber B., Winter S., 2002: Location Based Services using a Database Federation. In: Ruiz, M.; Gould, M.; Ramon, J. (Eds.), 5th AGILE Conference. Universitat de les Illes Balears, Palma, Spain, pp. 243-252.

[7] Jensen C.J., Christiensen A.F., Pedersen T. B., Pfoser D., Saltenis S. and Tryfona N., 2001: Location based services – A Database Perspective. In J.T Bjorke and H. Tveite (Eds). Proc. Of 8th Scandinavian Research Conference on Geographical Information Science, pp. 59-68.

[8] Lubinski A., 1998: Security Issues in Mobile Databases Access. In Proceedings IFIP WG 11.3 12th International Conference on Database Security

[9] Meissner A., Luckenbach T., Risse T., Kirste T., and Kirchner H., 2002: Design Challenges for an Integrated Disaster Management Communication and Information System. In the IEEE DIREN `02, The First IEEE Workshop on Disaster Recovery Networks.

[10] Müller J., 2006. Location based services Indoor Navigation. Presentation. ifgi.uni-muenster.de/~muellerj/lbs06/vortraege/8-IndoorNavigation.ppt

[11] Open Geospatial Consortium, http://www.opengeospatial.org/

[12] Persson, P., Espinoza, F., Fagerberg, P., Sandin, A., and Cöster, R., 2000: GeoNotes: A Location-based Information System for Public Spaces, in Höök, Benyon, and Munro (eds.) Readings in Social Navigation of Information Space, Springer.

[13] Rahman A. A. and Zlatanova S., 2006: Pre-Hospital Location Based Services (LBS) for emergency management In: E. Fendel, M. Rumor (Eds.); Proceedings of UDMS'06 Aalborg, pp. 11.49-11.57

[14] Zimmerman P., 1994: PGP User's Guide, The MIT Press.

[15] Zipf A., Malaka R., 2001: Developing Location Based Services for Tourism the service providers. In: P. Sheldon, K. Wöber, D. Fesenmaier (Eds.), Information and Communication Technologies in Tourism, Proceedings of ENTER 2001, 8th International Conference. Montreal, Springer Computer Science, Wien, NewYork, pp. 83–92.

[16] Zipf A., 2002: User Adaptive Maps for Location Based Services (LBS) for Tourism. In Proc. Conference for Information and Communication Technologies in Travel & Tourism (ENTER). Springer-Verlag.