



# A Location-aware Graphical BBS for Mobile Environments

Leichsenring, Germano

Sumiya, Kazutoshi

Uehara, Kuniaki

---

(Citation)

神戸大学都市安全研究センター研究報告, 4:167-176

(Issue Date)

2000-03

(Resource Type)

departmental bulletin paper

(Version)

Version of Record

(JaLCD0I)

<https://doi.org/10.24546/00231317>

(URL)

<https://hdl.handle.net/20.500.14094/00231317>



# A LOCATION-AWARE GRAPHICAL BBS FOR MOBILE ENVIRONMENTS

Germano Leichsenring <sup>1)</sup>  
Kazutoshi Sumiya <sup>2)</sup>  
Kuniaki Uehara <sup>3)</sup>

**Abstract:** We propose a graphical BBS which can handle the user's current location. In the system, each BBS comment has a valid time and an influence area decided automatically. The user's location then influences the system through events based on the user's movements to display comments on the user's screen. The influence area changes according to external events such as emergencies to express the importance and area of effect of the comment at the present time. Furthermore, we discuss the implement issues of the prototype system.

**Key words:** Mobile Databases, Location-Aware

## 1. INTRODUCTION

With the popularization of the Global Positioning System(GPS) and the Russian Glonass, GIS is playing an important role on mobile devices. Nowadays in Japan most cars use a Car Navigation System which can resolve routes and display information about shops and gas stations. Also, Personal Digital Assistants (PDAs) and mobile cell phones are extending their usability with a GPS receiver which can provide the user location for further processing.

The raw position information received from a GPS receiver is not very accurate. The sensibility is affected by buildings and can't be used indoors. Nevertheless, technology for increasing its accuracy on urban areas are rapidly evolving.

Nowadays systems access information from a static data source such as a CD-ROM or receive all information from a single source on the Internet. Although they show a high level of functionality, these systems can not accept information provided by ordinary users. The information to be provided to users has to be gathered and prepared by the organization or company that provides the service. Another problem with these systems is that the system becomes useless on emergencies. On an emergency, the information has to be fresh and location-related; mobile users need to be able to visualize information relevant to the place they are. It is well known that on emergencies such as earthquakes government offices unfortunately can't provide information quickly and accurately.

## 2. MOTIVATION

The World Wide Web and other Internet services became popular because, among other factors, ordinary users can provide information to others. Then, because of the excess of information provided, search engines flourished. With mobile computers and position receivers, we believe a similar system for location-aware information will be needed.

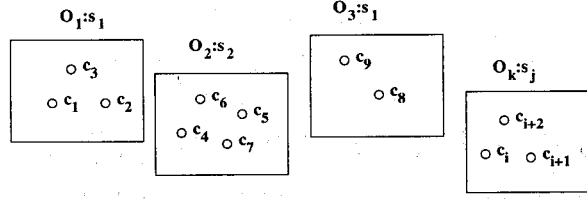


Figure 1: Comments included in Objects

An ideal system has the following requirements:

- Users can post information freely and easily;
- Users can visualize location-aware information easily;

For ordinary users to post information, methods to evaluate the importance of the information and avoid spam are needed. Then, other users can receive information related to their current position. This information should include a graphical visualization of the location the information is about.

### 3. OUR APPROACH

We define a *graphical Bulletin Board System* as a BBS which can handle graphics. A BBS room is identified by a real object which has a defined location on the map and a 3D model of the object. A comment consists of text about the room; by identifying the model with the real object the user can relate the comment with the place easier. We call a BBS room an *object* in this paper.

To display a comment, the client needs the 3D model of the object and the text of the comment. Here we suppose the user is walking, and thus the 3D models of objects near the user doesn't change very often and can be cached. Comments, on the other hand, are composed solely of text. These two factors enable the client to be a PDA connected to the Internet with a slow connection.

To evaluate the importance of comments, we use two factors assigned to each comment:

- influence area
- valid time

The influence area is the range a comment is supposed to affect. The valid time is a prediction of the freshness of a comment at the present time.

As the importance of each piece of information changes depending on the time, the influence area is dynamic; for example, if there is a room associated to a school, the range of effect of each comment varies when an earthquake occurred, when a festival is taking place or on a normal situation. We will say that these external events trigger *state changes* in this paper. The BBS objects, comments and their state is illustrated in Figure 1. In this figure each comment  $c_i$  is a comment about an object  $O$ . The positions are unrelated to the real position on the map.

To be able to send fresh comments to the user, each comment needs also a *valid time*, which is decided upon the comment's uniqueness on the object.

When a new comment arrives, its valid time and influence area are calculated; then, when an external event occurs, a state change is triggered and each of the affected comments have their influence area recalculated.

In the next section, we will explain how the system decides each of these two parameters and introduce our model.

Table 1: Sample Influence Area Function

Term	State 1	State 2
Nice View	20	-10
Free Food	0	30
Help	10	10
Tasty Food	5	0

#### 4. LOCATION-AWARE BBS MODEL

##### (1) Influence Area

The influence area of a comment expresses the comment's range of effect at the present time. As each object is related to a real-world object which has a defined location on the map, we use its center as the center for the influence area for every comment on the object. To decide the influence area radius, we use predefined terms and do a pattern matching on each comment affected by the state change; the terms found on the comment's text change the influence area radius according to an influence area function. For example, if we had two states:

**State 1** Normal Situation

**State 2** Emergency

we could have an influence area function such as the one in Table 1.

The system holds this set of predefined rules, and applies them when a state change occurs or when a new comment arrives through the following algorithm.

For each object  $O_k$  which is affected by the state  $s_j$ , the influence area radius  $I_i$  of a comment  $c_i$  belonging to object  $O_k$  is defined as follows:

$$I_i = I_j + \frac{\sum_{l=1}^m \text{area}(\text{term}_l, s_j)}{m}, \text{term}_l \in \text{text}(c_i) \quad (1)$$

where  $I_j$  is the default influence area radius for the state  $s_j$  and  $\text{term}_l$  is a term contained in the text of the comment  $c_i$ . As the presence of a predefined term indicates only a guess on the comment's content, we use the average of the predicted area for the terms found on the comment's text.

The influence area is illustrated in Figure 2. Here, each circle represents the influence area of a comment. For example, if an external event such as floods which triggers an urgent state( $s_2$ ) for the objects  $O_2$  and  $O_3$ , all comments  $c_4$  through  $c_9$  would have their influence area radius updated according to Table 1 and equation (1). It is important to note that the system does not change when an external event occurs; when there is an emergency which affects one or more objects, a state change on the objects triggers a recalculation of each comment's influence area and new comments on the object have the influence area decided according to the new state.

##### (2) Valid Time

For each comment the system needs a valid time, to be able to give the user fresh information. To achieve this we lower the time of expiration of old comments which are similar to newer comments. When an object has too many comments on the same subject, only the latest ones will be valid. The user, then, is able to receive the newest information on each subject.

When a new comment arrives, each earlier comment  $c_i$  on the same object have the valid time updated from  $v_{iold}$  to  $v_{inew}$  according to the following formula:

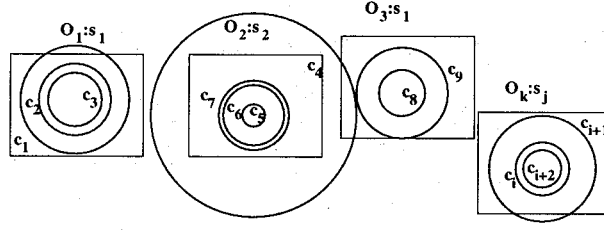


Figure 2: Influence Area

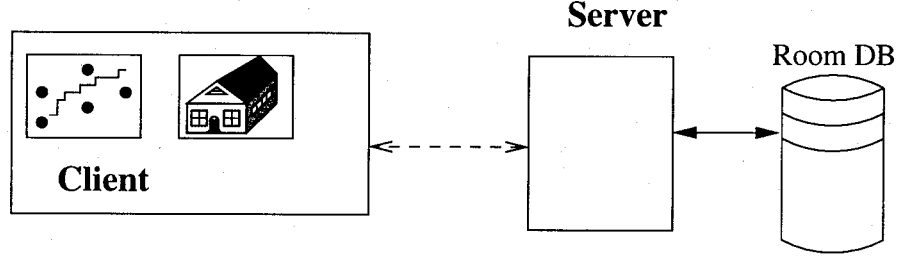


Figure 3: System Outline

$$v_{i_{new}} = \begin{cases} \text{sim}(c_i, c_{new}) \geq \alpha : & v_{i_{old}} * (1 - \text{sim}(c_i, c_{new})) \\ \text{sim}(c_i, c_{new}) < \alpha : & v_{i_{old}} \end{cases} \quad (2)$$

where  $\text{sim}(c_i, c_{new})$  is the similarity between the new comment  $c_{new}$  and a comment  $c_i$  and  $0 \leq \text{sim}() \leq 1$ .

### (3) System Architecture

Our idea turns out to be a Client-Server model. The client knows the user's current position, and the server, in turn, has all data related to an object and its comments. As we have a mobile environment in mind, the client has to hold also all information necessary to restart a connection. A system outline is shown in Figure 3.

Also, we expect the client to be able to connect to many BBS servers; this enables the user to choose servers according to personal preferences and geographical location.

#### a) Client Architecture

The client architecture is shown in Figure 4. Each client has a *display queue*, which points which comments should be displayed next. When a user enters a comment's influence area, the comment goes into the *display queue* only if the comment is not expired and the comment was not shown to this user yet.

When a user moves, the steps taken by the client to update the display queue are as follows:

- 1) *Event checker* receives the new position and asks the server for events.
- 2) *Event checker* then updates the *display queue* and notifies *comment downloader*.
- 3) *Comment downloader* downloads the comments on the *display queue* which will probably be displayed next.

*Display manager* pops each comment which is already downloaded and wasn't shown yet, shows them on the screen and updates the *shown comments table*.

The update of data on the client can take place every few minutes; the display manager, then, can show old comments again and again, until the connection is restarted. The display manager can also recheck the valid time and influence area of the comments downloaded and modify the display queue dynamically. Unless the user moves fast, the only drawback of not having a constant connection is in the event of disasters, when the influence area of comments are changed on the server and the client database can become useless in some situations.

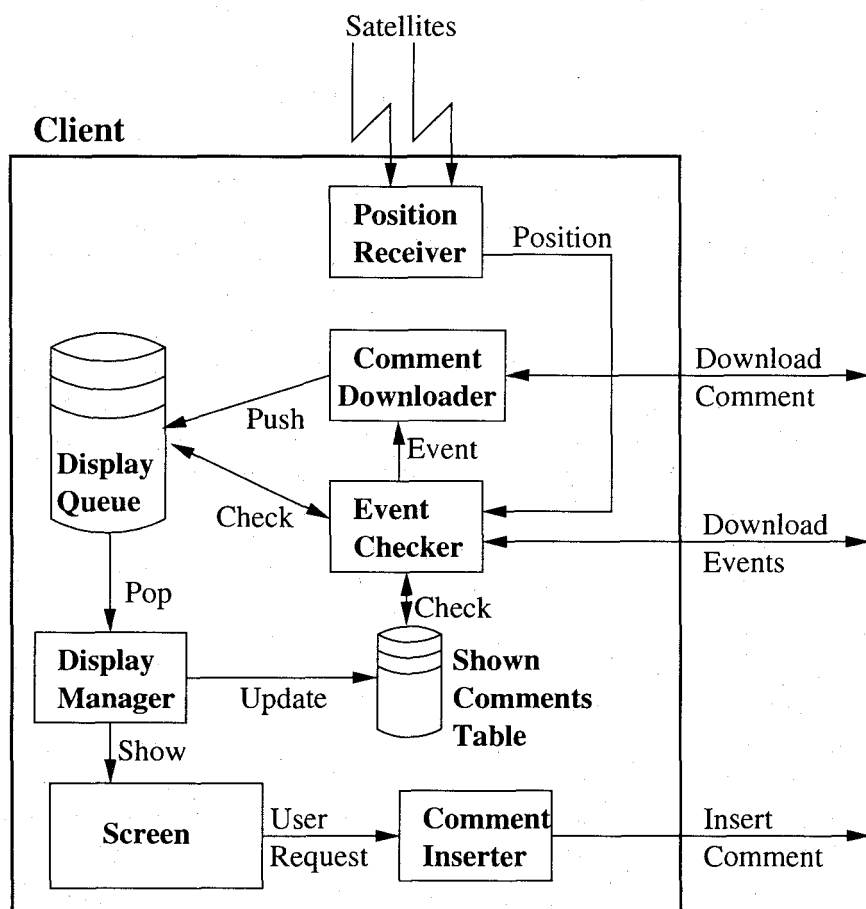


Figure 4: Client Architecture

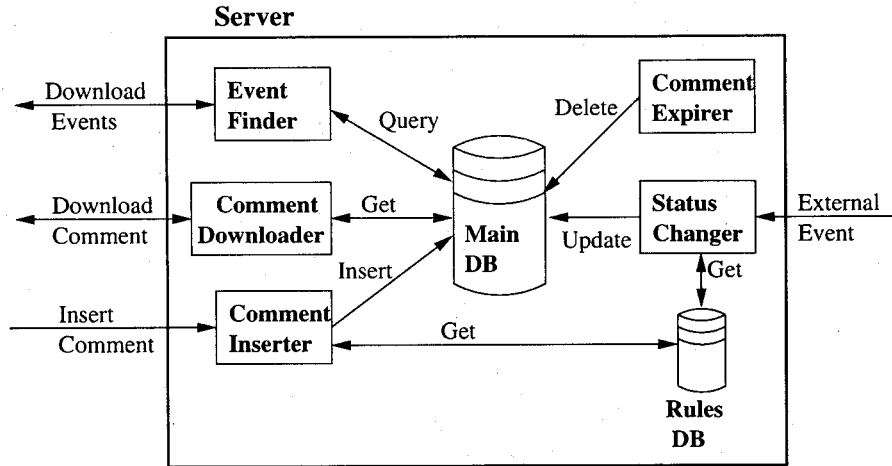


Figure 5: Server Architecture

## b) Server Architecture

As illustrated in Figure 5, the server has several modules almost unrelated to each other with the exception of being connected to common databases. The *event finder* searches for comments which influence area contains the user's position and are not expired. The *comment downloader* downloads comments. The *comment inserter* inserts comments, calculating the valid time according to equation (2) and the influence area according to equation (1). The *comment expirer* checks for comments which valid time has passed and expires them.

The *status changer* waits for external events; when an external event occurs, it changes the state  $s_j$  of the affected objects through the following loops:

- 1) for each object  $O_k$  affected by the external event  $s_j$
- 2) for each comment  $c_i$  contained by the object  $O_k$
- 3) update the influence area  $I_i$  according to equation (1).

## 5. PROTOTYPE SYSTEM

### (1) Implementation

Many variables of the model can have an impact on the efficiency, speed and usability of the system. The most important one, the influence area function, is particularly hard to estimate for different situations. With a prototype system we plan to, on a first step, evaluate the system on normal situations, and on a second step, simulate the usage on disaster situations with information gathered about the Kobe Earthquake. The prototype system put a low load on the client so that it can be ported to PDAs in the near future.

Our prototype system uses a plane coordinate system, which enables fast nearness calculations but limits the usability of the system on a large area because of the approximation. The server is developed as Perl CGI scripts which access a PostgreSQL database. The user position is received by the event finder which queries the database for events. To award an influence area to each comment, we do a text search for terms on a database. The *comment expirer* is implemented as a surveillance process on the server which looks for comments which have gone beyond their valid time. Instead of a 3D model we use a single picture of the object, so the direction is not considered.

To insert comments, the user accesses a WWW page generated by the comment inserter perl script. The other parts of the client is implemented as a Java2 program which connects through HTTP to the server. It can connect to multiple servers and display the comments not shown yet on an influence area order, updating automatically every few seconds. These

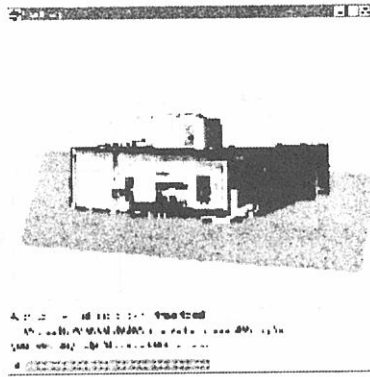


Figure 6: Client Window

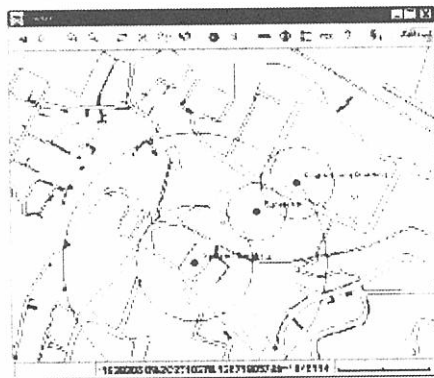


Figure 7: JaMaPS Window

comments are cached so an interruption on the connection doesn't affect the displaying. The position is acknowledged through a GPS receiver.

The client is composed of two windows:

**Object 3D viewer** shows a picture (Figure 6, upper side)

**Comment viewer** shows an HTML page (Figure 6, lower side)

For better performance, the client downloads the picture separately from the comment and caches them.

Although not part of the model, there is a need to display a map so the user can visualize the real position and check if the results are the expected ones. We are presently using JaMaPS<sup>1)</sup>, a client-server system developed by KDD where the server sends maps to the client on demand. The JaMaPS client window is shown in Figure 7. Here the dark dots are the BBS objects; the circles are the influence area of comments on the object; and the shadowed dot is the user's current position. As at the present time there is no way to input the user's position directly to the JaMaPS client, we opted to put the IP address and real position of the user on the server database and use it to generate a dynamic map for the JaMaPS client.

## (2) Experimental Environment

Figure 8 is a simplified map of the Kobe University campus. Every building with an object related to it has a point which is the center for the influence area of comments posted on the object. The objects are as follows:

$O_1$ : Front Building

$O_2$ : Computers Building



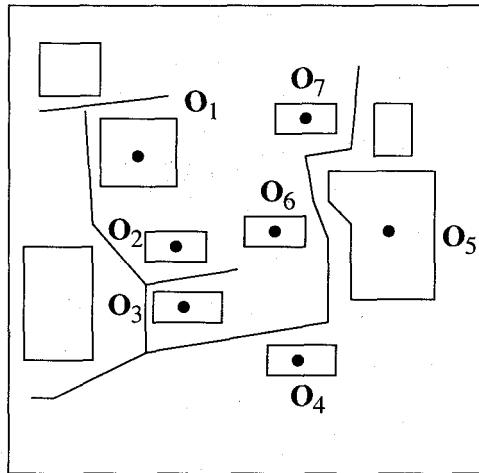


Figure 8: Campus Map

$O_3$ : Systems Engineering Building

$O_4$ : Cafeteria-1

$O_5$ : Engineering Building

$O_6$ : Cafeteria-2

$O_7$ : Computers Building

Making real world experiments at Kobe University we plan to find good terms which can be used for the influence area function.

### (3) Usage

#### a) Information Retrieval Example

The steps taken to update the comments and 3D models on the client are as follows:

1. The user's position is acquired
2. The client asks the server for IDs of comments relevant to the location of the user
3. The client downloads comments and 3D models if not on the local cache according to importance; higher importance comments first
4. All comments go to the display queue

The display manager, shown in Figure 4, recheck the position to see if the user is still inside the comment's influence area and the comment is still valid and display these cached comments in order to the user. When the client is disconnected, cached comments are used and when the position can not be retrieved because of interference or other problems, the last position is used.

#### b) Information Posting Example

When a user is looking at comments about a BBS room, the user can post extra information on the room. The comment is sent to the server and then processed as follows:

1. The state of the room and the rules for the state are retrieved from the database
2. The rules are applied to the comment and the influence area radius is calculated
3. The comment is awarded a default valid time
4. All earlier comments on the room still valid have their valid time updated

## 6. RELATED WORK

Several research efforts have focused on providing location-aware 2-D and 3-D media for a mobile client. HyperCampus<sup>3)</sup> is a system for providing location-aware data according to the user's preferences through agents. At Columbia University<sup>4)</sup> a system for providing multiple types of location-aware media to the user have been developed. Both systems focus on the

location-aware problem but don't take into account user's comments. The Columbia University system's interface consists of a head orientation tracker and a head-worn display, which together can provide an augmented reality to the user. This interface can be trivial on disasters if combined with our system. The user could then understand what is happening around her in a glance.

Arikawa<sup>5)</sup> developed methods of using the location data embedded on video frames to browse and retrieve video data spatially; the method can't, at the present time, be used on mobile environments. Banjou et. al.<sup>6)</sup> developed a system for retrieving and browsing pictures depending not on the location of the photographer but the desired object's location.

Tarumi et. al.<sup>7)</sup> developed a system for deploying location-aware tags (SpaceTags) to users. SpaceTags can be interesting from a business or entertainment point of view, but the lack of methods for automatically calculating the influence area and valid time and changing the system's state does not fit our purpose.

## 7. CONCLUDING REMARKS AND FUTURE WORK

We have presented a BBS model which has the ability to provide fresh, location- and time-aware information to its users. The system is based on a valid time determined upon the comment's uniqueness and an influence area determined upon the comment's expected importance and range of effect at the present time. The system relies on BBS rooms related to real world objects. Although GPS receivers can't be used indoors, the location information needed for this system doesn't have to be accurate. Furthermore, our system can be deployed with nowadays existing technology (GPS-extended PDAs) and infrastructure (the Internet and cellular phones).

We expect the BBS rooms to be far from each other; there are two issues not addressed in this paper which have to be considered when rooms are near each other. The first one is the performance issue; many 3D models have to be cached for better performance, but on the other hand the storage capacity of mobile devices is limited. The other one is that the calculation of the freshness of each comment should take into consideration comments on other rooms as well.

The influence area function affects strongly the reliability of the system. Experiments and research are needed to find improved methods for setting the influence area.

## REFERENCES

- 1) Satoru Takagi, Arei Kobayashi, Kenichi Asakawa, *JaMaPS Distributed Digital Mapping Platform Based on Web*, IPSJ SIG notes 99-ITS (in Japanese)
- 2) W3C Scalable Vector Graphics <http://www.w3.org/Graphics/SVG/>
- 3) Katashi Nagao, *Mobile Computing and Agent Augmented Reality*, Journal of Japanese Society for Artificial Intelligence Vol. 14 No. 4 pp 579-589 (in Japanese)
- 4) Tobias Hollerer, Steven Feiner, John Pavlik, *Situated Documentaries: Embedding Multimedia Presentations in the Real World*, In Prof. of 3rd International Symposium on Wearable Computers
- 5) Masatoshi Arikawa, *Spatial Browsing for Video Data Using Spatio-Temporal Descriptions of Cameras*, IPSJ SIG Notes 98-DBS-116 pp 225-232
- 6) Yutaka Banjou, Hiroki Takakura, Yahiko Kambayashi, *Visual User Interface for Spatial Databases Combined with Landscape Images and Sensors*, The Institute of Electronics, Information and Communication Engineers (IEICE99) (in Japanese)

- 7) Hiroyuki Tarumi, Ken Morishita, Megumi Nakao, Yahiko Kambayashi, *Space Tag: An Overlaid Virtual System and its Applications*, Proc. International Conference on Multimedia Computing and Systems (ICMS99) pp 207-212

**Authors:** 1) Germano Leichsenring, Graduate Student, Graduate School of Sciences and Technology, Kobe University; 2) Kazutoshi Sumiya, Lecturer, Research Center for Urban Safety and Security, Kobe University; 3) Kuniaki Uehara, Professor, Research Center for Urban Safety and Security, Kobe University.