Mobile Augmented Reality System for Interacting with Ubiquitous Information

Andriamasinoro Rahajaniaina Jean-Pierre Jessel
Institut de Recherche en Informatique de Toulouse
Université de Toulouse – Paul Sabatier
Toulouse, France
{Andriamasinoro.Rahajaniaina, Jean-Pierre.Jessel}@irit.fr

Abstract—**In this paper we describe a mobile collaborative Augmented Reality system which allows multiple users to visualize and to interact with a distributed parcel's information in a mobile device using wireless network. This information augmented the live images of the real environment surrounding the user. The main goal of our work is to use the replication of the message dispatcher server with message filter in order to allow each client allowing load balancing between them and to permit the message dispatcher server dispatch all updates messages to the concerned clients only. Our system explores a collaborative Augmented Reality application across several hardware device, and multi-user interface adaptation.**

Keywords - Ubiquitous collaborative computing; mobile collaborative augmented reality; distributed Geographical Information System.

I. INTRODUCTION

Augmented Reality (AR) is among the more popular techniques used to perform user interface for ubiquitous applications. AR presents information in its context within a 3D mixed environment.

The problem of mobile AR is the diversity of the mobile devices (visualization and interaction). In this case tools for visualizations and interaction must take account of this constrained. Most of the previous researches [1][2][3] that used Geographical Information System (GIS) database relating for building, streets in order to enhance the experiences of users (e.g., tourists, visitors). These previous works are dedicated for one or few platforms and applications. Instead, the system presented here tries to surmount this problem. We propose a collaborative augmented reality system to visualize and change the parcel's characteristic depending on location and context. As GIS database stores numerous data, so retrieving information from it is among of one critical point in a distributed AR system because it may generate latency during exchange. To overcome this problem, we show in detail our approach about retrieving information related to a Parcel from the GIS database, and the adequate metaphor for visualization and for interaction.

Outdoor AR systems have traditionally been reserved to use GIS databases related to buildings and streets in order to provide help to users.

First, the Mobile Augmented Reality Systems (MARS) project [1] allowed user to arrange the multimedia information according to chronological order. This system used a campus database to overlay labels on buildings seen through a tracked head-worn display.

The Archeoguide project [4] was designed to increase real images of user's environment with virtual story information related to them.

Next, in [5], the authors presented a prototype of an interactive visualization framework specifically designed for presenting geographical information in both indoor and outdoor environments. Participants can visualize 3D reconstructions of geographical information in real-time.

Then, ARscouting system [3] allows the mobile client (scout) takes several images for instance of a target building and transmitted them to a custom database. After that, the reconstruction engine gets a notification and triggers the reconstruction process. Once the reconstruction task is over, the server stores the virtual object and transmits it to the scout in order to increase user interface.

The claimed Mobile Augmented Reality Applications (MARA) [2] allows users to interact with their surrounding environment using the standard mobile device inputs. The users could share or exchange all data with others connected users.

This paper is organized as follows. Section 2 presents a detailed description of our *ARGisUbiq* system. After that, we give experimentation and result in Section 3. Section 4 presents conclusion and future works.

II. ARGISUBIQ SYSTEM

ARGisUbiq system is an improved version of our preview work [6]. The main goal of *ARGisUbiq* system is to propose a collaborative application AR GIS in agronomic domain that shows all information about a parcel according to the user's location.

The main difference between this work and preview ones is the use of the message distributed server for managing the collaboration between clients and to maintain the coherence of the database on all the clients and servers.

A. Message dispatcher server

This subcomponent is dedicated to dispatch all messages between the clients and the servers. It uses IceStorm service from Internet Communication Engine (ICE) [7] and applies publisher and subscriber principles. As IceStorm allows the replication of the message dispatcher via IceGrid, we create three identical ones in order to avoid that the message dispatcher become a bottleneck. In this case, the publisher could make a load balancing between these dispatchers of messages. Each dispatcher has a topic manager which manages one or more topics activities. According to the client's number, it is possible adding a new message dispatcher dynamically by changing the server's configuration file.

Each topic can have several publishers and subscribers. In this scheme of communication, the slave servers and clients are the subscribers and they must subscribe on a topic in order to receive all message from this one. A topic is identified by its name in the gateway and it is the responsible of the message dispatching deposited by the publishers to all subscribers on this topic.

As all Remote Protocol Communiction (RPC), a publisher has a proxy of the publisher on the server in order to deposit its message to the topic. When receiving the message, the servant on the topic dispatches it to its subscribers.

a) Type of message: With aim of having the best exchange, we use two kinds of messages: update messages and synchronization messages.

An update message is deposited by a publisher after an user interaction which changes the state of the additional object in the augmented space. This message will be dispatched to the subscribers.

A synchronization message is distributed in order to maintain the coherence of the data on all clients and servers, the master server deposits a synchronization message on the dispatcher of message in a regular time interval. This one contains all last updates and which will distributed towards in all participants including the slave servers.

With this method of communication, the distributed messages are occupied the network's bandwidth when several publishers deposited their messages in the same time and generated a problem. As ICE does not envisage this kind of problem, we add our message filter on the distributed message server in order to distribute this one only to the concerned subscribers. We describe this technique in the following section.

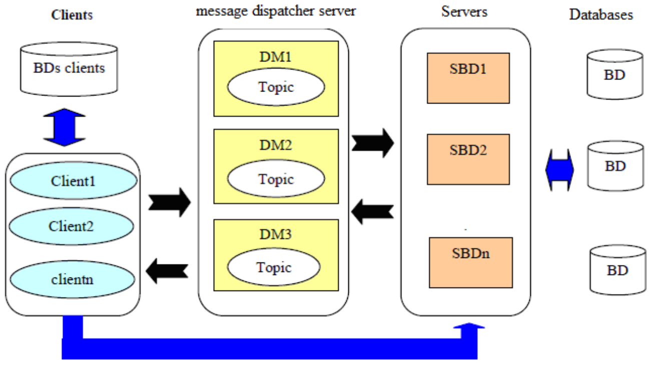


Figure 1. The general architecture of communication

b) Message filter: We have two possibilities to install this filter: at the time of the creation of the topic or in the servant. In the first case, it is enough to set up a rule or naming the topic. In other word, the topic's name will be followed by a suffix whose value is the bounding box where the creator of topic is located. Consequently, the messages will be distributed only to the subscribers which are in this bounding box. The disadvantage here is that the publisher should create a new topic with each time it change a bounding box. It can involve a waste of time since it should disconnect from the old topic and be established a connection to the new topic. This waste of time will have an impact on the fluidity of the user's interaction.

For the second possibility, we use only one topic and the value of the bounding box is deposited by the publisher with its message. After that, the servant will be distributed this one only to the subscribers which have the same value of bounding box. In this case, the publishers do not need to change topic with each time they change a bounding box. We chose this second type of filter in order to have more fluid interaction. This one is useful only for the update messages because only the users close to the initiator of change will be interested. We use message dispatcher server to manage the user's collaboration which we will see in the section below.

B. *User's collaboration*

This work allows users to collaborate. The main goal is to manage the update of the juridical situation of a parcel. Each participant could change this one only in *hybridview* mode and the flag value is equal to zero. On the other hand, the other users are not obliged in *hybridview* mode. Once the process of modification is engaged, the flag value becomes to 1 (parcel locked).

The type of collaboration depends on the location of each participant with respect to the initiator of modification one:

1) Synchronous collaboration: this mode of collaboration proceeds between initiator of modification and the other participants who are in the same bounding box. Indeed, the message of updated juridical situation is immediately sent to these participants only in order to update in real-time their data and their interfaces of visualization (see Figure 2).

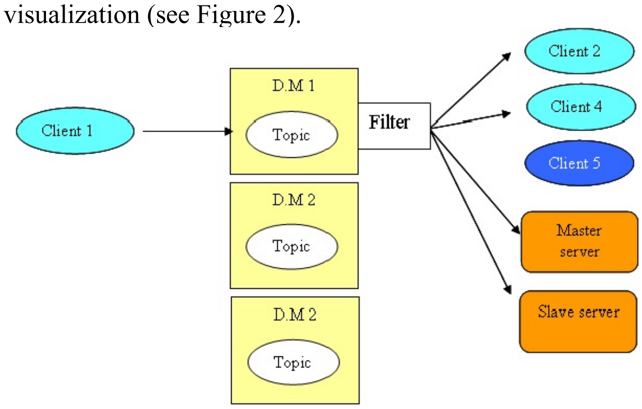


Figure 2. The update message dispatching

2) *Asynchronous collaboration:* this case arises between initiator of the modification and the others participants who are located in other bounding box. In this case, modifications are distributed asynchronously to these participants in order to update their databases. This principle is adopted because these ones are not directly concerned with the modification. During the modification, the other users cannot make modification.

C. Selection Information source

Like others outdoor AR systems, *ARGisUbiq* system uses GIS data as data source. As parcel's information related to agronomic and type of plant is unavailable on producer's map, we create our own database inspired from parcel database (using a vector format formed by shape files, index one and dbf one).

Each polygon in the file shape is delimited by a bounding box. We exploited these limit in order to know the parcel in which the operator is located. Thus, we tested two methods: first is based on the calculation of the barycenter and the second is called Winding number.

1) *The barycenter method:* This method is used especially when there are several parcels inside the bounding box. We calculate the barycenter of each parcel. We take account only the parcel having its barycenter nearby the user's position. Once the found parcel, one will recover other information by using the index of the parcel. The problem with this method, it is that the operator can be inside the bounding box but it is not always inside a parcel because the fact of having the smallest distance compared to the barycenter does not signify that it is inside. To solve this problem, we use the Winding number method which we will see in the section below.

2) *The winding number method:* As the parcels stored in the shp file format (shape file) are convex polygons, then we can apply the Winding Number method [8].

This method makes possible to know the inclusion of a point in a convex polygon. The algorithm is summarized as follows: we take an infinity ray R starting at a point P and we calculate the number of intersection where R crosses the segments which form the polygon. If R crosses a segment of the polygon whose has the clockwise orientation, the counter value is equal to -1. On the contrary, it is equal to +1.

We do not take account of the segments parallel with R. If the sum of these numbers is equal zero that means that the point P is outside the polygon. In contrast, it is inside.

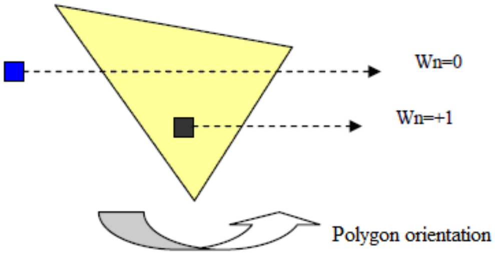


Figure 3. The application of Winding number method

By replacing the point P by the position of the operator and the polygon by one of the parcels in the database, we applied this algorithm after having tested the inclusion of the position of the operator in the bounding box. This test of inclusion in the bounding box enables us to avoid the useless execution of the algorithm for the points outside the bounding box. When we found the parcel, we will recover other information by usin*g the index of the parcel.*

D. Visualization metaphor

To visualize the GIS information selected as described in the preview section, we proposed two metaphors of visualization: the metaphor of visualization in *textview* mode and the metaphor of visualization in *hybridview* mode. We added audio augmentation to these metaphors not to obstruct the sight of the operator on the one hand and on the other hand in order to provide the details of information by overcoming the limits imposed by the size of the screen.

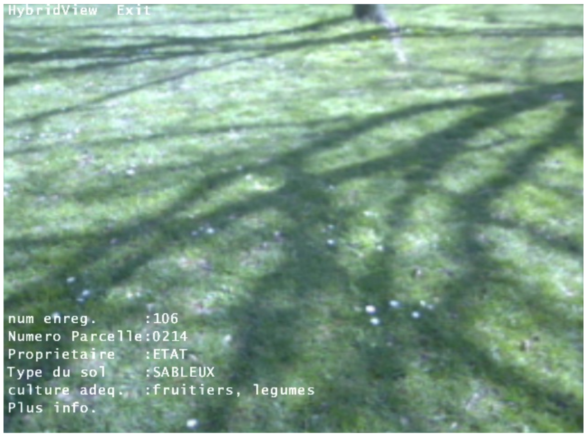
In this case, additional information essential to the sight is used for the visual augmentation and the others are dedicated to the audio augmentation.

The combination of these various modes of augmentation enables us to make a multimodal augmentation instead of limiting to the usual visual one. This one offers us opportunity of providing an identical metaphor of visualization for the various types of used devices.

In the *textview* mode, the scene is augmented by virtual text and aural information related to a parcel and the position of user. In the *hybridview* metaphor, user interface is enhanced by virtual text, parcel map and audio information.

The blue point on the map is the user's position. We combine landscape and portrait mode with *textview* and *hybridview* when users use lightweight hardware as visualization tools.

The transition between the two metaphors depends on the way which the user holds his PDA. The *hybridview* is the default visualization metaphor for UMPC and Personal Computer (PC) notebook, and the *textview* metaphor is for PDA. Each juridical situation is associated with a color of which green for *public property*, yellow for *public property (request in progress)* and red for the juridical situation *private property*.



```
num enreg.      :106
Numero Parcelle:0214
Proprietaire    :ETAT
Type du sol     :SABLEUX
culture adeq.   :fruitiers, legumes
Plus info.
```

Figure 4. The textview mode on an UMPC

We decide to use classical interaction tools like menu, stylus, and button because these are available on each device that we use as visualization tools. When the user selects one of both menus using his stylus, the user interface changes according to the menu item selected. After that, the menu item changes to another one.

E. Interaction metaphor

We propose the possibility for user to choose visualization metaphor using *textview* and *hybridview* menus. As described above, when using a Personal Digital Assistance (PDA), the transition between the two metaphors depends on how the user holds his PDA and the value of pitch angle (Θ) from the inertial sensor Xsens MTi (Θ value between -5.0° and 0.0° for portrait mode and landscape mode for others values) see Figure 5.

When the current mode of visualization is the *hybridview* mode, the operator can modify the juridical situation of the parcel visited while clicking on the map. According to the situation of the parcel, it can pass from the juridical situation *public property* to the situation *private property* while passing by the intermediate situation *public property (request in progress)*. Only the parcel in *public property* or *public property (request in progress)* situation accepts the modifications of juridical situation (see Figure 6).

To manage the user's collaboration, we attribute for each parcel a flag which designate its modification state. An operator can change the juridical situation of a parcel only if its flag value is equal zero. When a parcel is in an intermediate phase (an operator interacts with it to change its state) the flag value becomes equal to 1.

III. EXPERIMENTATION AND RESULTS

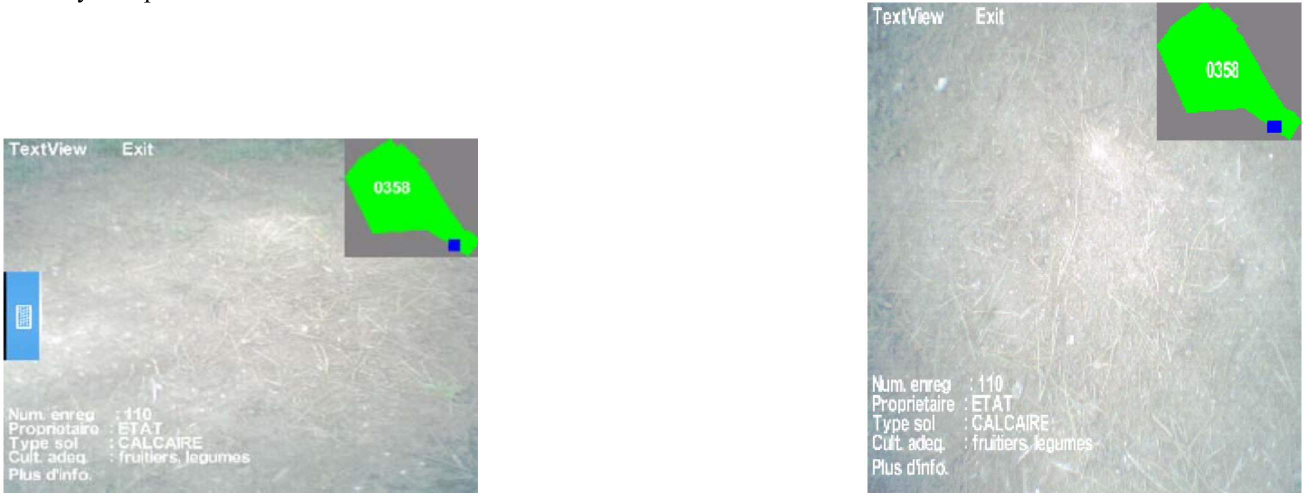
This first prototype was tested with three platforms: the first client has used Q1 Samsung with 800 Mhz Celeron M ULV processor, 256 Mo RAM and the two other clients have used a Pocket PC Dell Axim x51v with 624Mhz Intel Xscale processor, 64Mo RAM. We have used a database of common formed by 180,000 parcels and each parcel is formed by 10 up to 20 vertices. We have tested two different scenarios: first, we have used a local database: as we have loaded the database in the memory at the first time, the Q1 client has run after 5 s of the database loading and 22 s for the two PDA clients. They do not need to establish a connection to the database server but they must to connect to the dispatcher of the message server. This last one is formed by three dispatcher of message running on a Linux Ubuntu operating system. This connection has spent 0.3 s. After this step, the Q1 client was able to achieve 25-30 fps (frame per second) and 17-20 fps for the PDA clients during the exchange with the data in memory.

In the second test, the database is duplicated on three replica servers: one master registry and two slave ones. Each slave registry has had its own node which monitors two applications servers. The locate request from the client to the registry has spent 0.3 s for Q1 client and 0.7 s for PDA clients. After that, the Q1 client was able to 23-28 fps and 15–20 fps for the PDA clients during the exchange with the replica server. As describe above, all update message must deposit on the dispatcher of the message server.

As we saw, the difference between the results using the local database and replicate database was small. It's not surprising because we have added to the client a functionality to reduce the number of exchange with the replica server. It's also due to the performance of our replica servers: 50% of users only are satisfied for hardware device ergonomic because most of users prefer using wireless inertial sensor instead of using MTi.

As we describe above, our work could overcome the network problem because we are working in memory to reduce the exchange with the database servers. Consequently, our system permits the users to collaborate in real-time. Moreover, it does not consume more memory time.

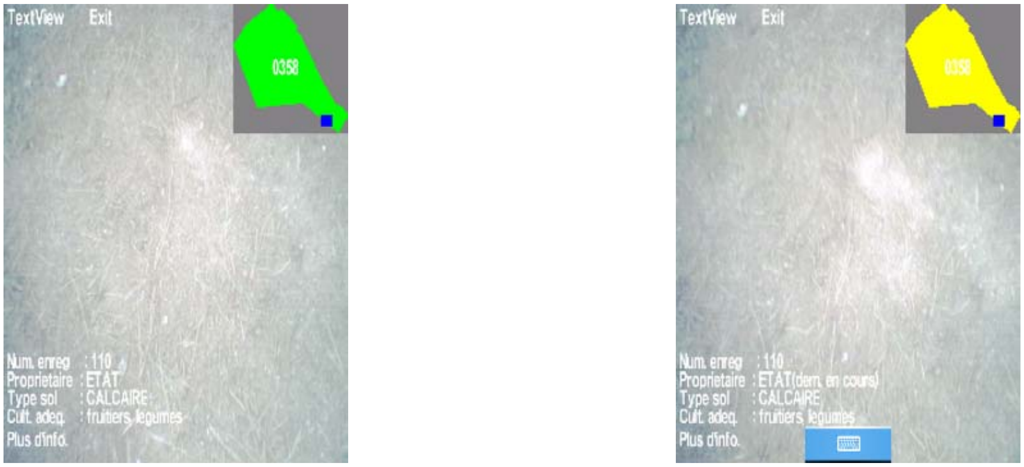
From the experimentation in the long run, the pitch values have presented any drift but it has not impact to our system because this one does not need exact pitch values for running. In other hand, the pitch values must be in the interval as we describe in [6].



Num. enreg : 110
Proprietaire : ETAT
Type sol : CALCAIRE
Cult. adeq. : fruitiers, legumes
Plus d'info.

Num. enreg : 110
Proprietaire : ETAT
Type sol : CALCAIRE
Cult. adeq. : fruitiers, legumes
Plus d'info.

Figure 5. The landscape hybridview mode (left) and the portrait hybridview mode (right)



Num. enreg : 110
Proprietaire : ETAT
Type sol : CALCAIRE
Cult. adeq. : fruitiers, legumes
Plus d'info.

Num. enreg : 110
Proprietaire : ETAT(dem. en cours)
Type sol : CALCAIRE
Cult. adeq. : fruitiers, legumes
Plus d'info.

Figure 6. The portrait hybridview mode before change (left) and after change (right)

From these results we can deduce design guidelines to choose hardware device in future AR applications.

IV. CONCLUSION AND FUTURE WORK

In this paper, we addressed the problem of enhancing user's contextual perception of the real word using GIS data on several hardware and software platform. To tackle this, we have proposed the *ARGISUbiq* multiplatform architecture which exploits Mobile collaborative Augmented Reality principles to improve user's interaction with GIS data. As we use specifically built GIS data, we described our database's structure and how to select appropriate information related to user's position. Our distributed application is based on Internet Communication Engine, an object-oriented middleware, used to ensure the connection between database servers and clients. To avoid eventual problem with database server, we duplicate our database on several servers and we use IceGrid services to provide load balancing between all servers. Some clients are able to access concurrently to a selected server.

We use IceStorm services on the message dispatcher server to manage the dispatching of the update and synchronization message to the subscribers and the slave servers.

We are entirely satisfied with our results. In the future work, instead using MTi sensor we plan to use low cost or embedded inertial sensor and computer vision based techniques to compute the user's orientation.

REFERENCES

[1] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway, "Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System", Computers and Graphics, 23(6), Elsevier Publishers, 1999, pp. 779-785.

[2] M. Kähäri and D. J. Murphy, "MARA-Sensor based Augmented Reality System for Mobile Imaging Device", http://research.nokia.com/projects/MARA/, 18 July 2010.

[3] B. Reitinger, C. Zach, and D. Schmalstieg, "Augmented RealityScouting for interactive 3D reconstruction," unpublished.

[4] T. Gleue and P. Daehne, Augmented Reality-based Cultural Heritage On-site Guide, http://archeoguide.intranet.gr, 18 July 2010.

[5] F. Liarokapis, I. Greatbatch, D. Mountain, A. Gunesh, V. Brujic-Okretic, and J. Raper, "Mobile Augmented Reality techniques for GeoVisualisation," Proc. 9th International Conference on Information Visualisation, IEEE Computer Society, London, 2005, pp. 745-751.

[6] A. Rahajaniaina and J. Jessel, "Visualization of Distributed Parcel's Information on Mobile Device," Proc. International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2010), St. Maarten, Netherlands Antilles, IEEE Computer Society, February 2010, pp. 27-32.

[7] M. Henning, M. Spruiell, D. Boone, B. Eagles, B. Foucher, M. Laukien, M. Newhook, and B. Normier, Distributed Programming with Ice, ZeroC: http://www.zeroc.com/Ice-Manual.pdf, 18 July 2010.

[8] D. Sunday, Fast Winding Number Test for Point Inclusion in a Polygon. http://softsurfer.com/algorithms.htm, 18 July 2010.

ISBN: 978-1-61208-100-7