Proceedings of TeleGeo'2000: Second International Workshop on Telegeoprocessing. 2000, p. 53-61.

Use Cases of vehicle location systems based on distributed real-time GPS data[1]

F.J. Zarazaga, P. J. Álvarez, J. Guillo, R. López, J. Valiño, P.R. Muro-Medrano

Computer Science and System Engineering Department
University of Zaragoza
María de Luna 3
50015 Zaragoza, SPAIN
{javy,juanv,prmuro}@posta.unizar.es
{alvaper,jguillo,lrafa}@ebro.cps.unizar.es
http://iaaa.cps.unizar.es

Abstract. In modern computer environments used for vehicle location several users, with different technical or managerial goals, may be interested in the real time data coming from the GPS sensors embedded in the vehicle or may want direct access of the GPS system itself. On the other hand, the connection to these remote GPS devices may be also strategically distributed along different radio communication sites because of radio coverage reasons. This work describes a CORBA based system architecture used to obtain those capabilities and provides some understanding on real possibilities for its practical use. These possibilities are shown through three use cases of vehicle location systems, offering architectural solutions that allow flexible and easily scalable deployments.

Keywords. vehicle location systems, distributed systems, real-time GPS, CORBA

Introduction

Real-time GPS data [6] is the most common information source for vehicle location systems (some examples of them could be find in [1], [2], [5] or [7]). Usually, this kind of systems has a component to collect the GPS data from the remote devices. This component is installed in one computer connected with one o more radio terminals (with the same or different technology). The remainder of the system accesses this component from the same or different computer (connected to it through a network). This architecture could have some limitations in some aspects such as scalability (the acquisition PC is the bottleneck of the system in situations of high radio communications level) and robustness (problems in the acquisition PC have a high and quickly impact in the operational capacity of the whole system). Furthermore, there could be some problems related with radio technologies. Most of them have radio coverage problems because they manage the communications in different areas not related (for example GSM across different countries).

The use of more than one GPS data collector, distributed transparently along any kind of network, could offer new possibilities to manage all this kind of problems. Moreover, it implies the use of a new architecture which can offer new possibilities to build complex vehicle location systems in an easier manner. In this paper, we want to present some of these possibilities by showing three use cases of applications which work with distributed real-time GPS data collectors. The deployment of these applications shows that they are very modular and easy to scale. Many of the components used in one system are replicated across the different installations.

These applications are based in OODISMAL [8]. This is an object oriented distributed information system for mobile automatic location that provides the basic components to integrate radio communications, real-time data captured by GPS units, on-line and off-line position analysis, and GIS components.

The paper is organized as follows. In section 2 the basic system is presented. Its architecture is accompanied by two-sample deployment: laboratory test bed and the stand presentation in a professional fair in France. Section 3 presents the three use cases: snowploughs tracking, ambulances network management, and a monitoring and user information system for small cities urban buses. This work ends with a conclusion section.

[1] **Acknowledges**: The basic technology of this work was partially supported by the *Comisión Interministerial de Ciencia y Tecnología* (CICYT) of Spain through the project TIC98-0587 and by the *Consejo Superior de Investigación y Desarrollo* (CONSI+D) of Aragón through the project P-18/96.

System Overview

Architecture

OODISMAL (see **Fig. 1.**) is an information system that is composed of a set of distributed components [14] in a LAN (Local Area Network). Distributed Component Software technology makes possible the development of light applications where it is solved the specific problematic of the application, whereas the basic functionality is provided by reusable components. A distributed client-server application, where clients and servers may be resident in the same or different machines, makes possible to increase the system functionality without the modification of reused parts. Servers are used only when it is necessary, and may be shared by several light remote or local clients.

Components developed in OODISMAL interoperate between them using the CORBA infrastructure [9]. This technology allows the use of more than one instance of the same server working together. It is very useful when it is necessary to offer distributed services as central services (like the use of more than one GPS data collector), to give access to different devices with the same interface (like the use of different communications technologies at the same time), or to balance workflow (adding new machines to the net with new instances of the server). in this way, it is possible to increase the global functionality of the system or to replace server with new version without rebooting the rest of the components, because server access is solved in run time.

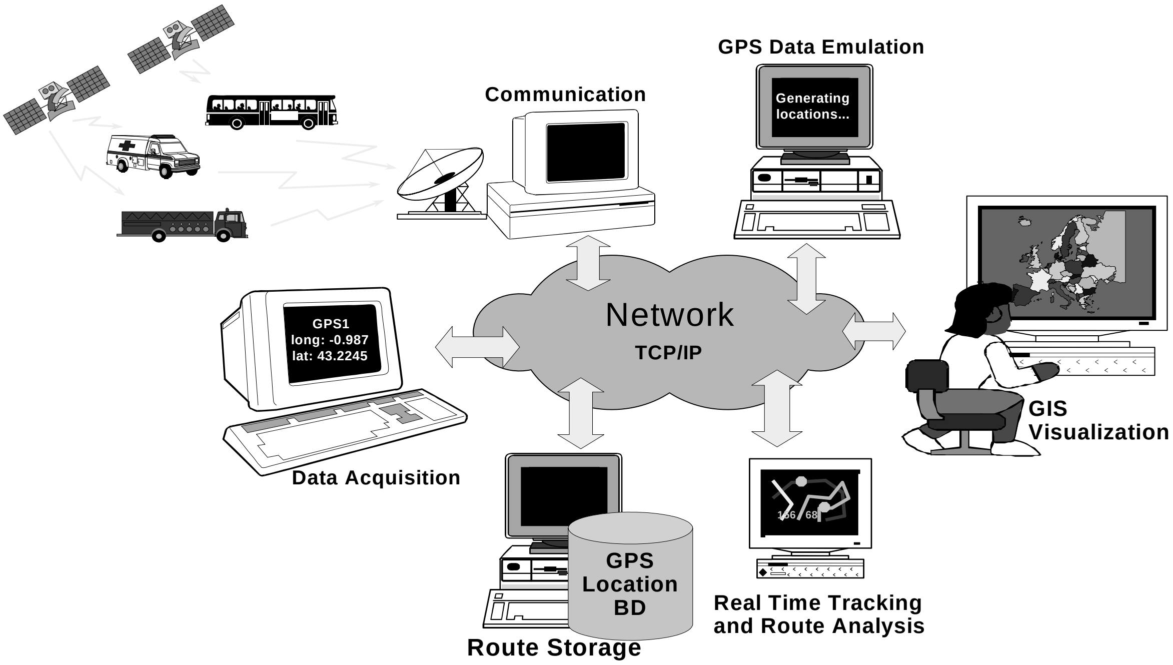


Fig. 1. OODISMAL architecture.

Basically, OODISMAL offers a library of components with the following characteristics:

Flexibility. Components can be replaced with new versions without impact on the remainder of the system.

Reusability: Same components can be used for different kind of applications.

Scalability: Easy to adjust to the final client necessity (local or distributed; more than one component of the same type working at the same time, if needed).

Robustness. It is possible to reduce the risk of component or PC crash by duplicating them in different PCs.

The main components presented are the following:

Remote devices. In order to track the location of remote vehicles or people, some kind of device is needed in the remote location. It typically consists of a GPS receptor and a communication equipment, such as a mobile telephone or a radio device. These two devices may be connected directly, and may even be integrated in a single one. If not, they can be put to work together by connecting both of them to a laptop or palmtop computer.

Communication component, which allows the system to communicate with remote GPS devices using wireless media.

Data Acquisition component, which builds and decodes messages for the remote GPS devices, including location queries, location responses and GPS configuration. It uses the Communication component to send and receive these messages.

Real Time Tracking component, which acts as a client for the Data Acquisition component. It gets the locations of the remote devices and processes them as needed. For example, it can be connected to GIS Visualization or Route Storage components.

GIS Visualization component, which shows locations and routes on digital maps. Besides, it offers the typical map management functions (zoom in, zoom out, pan, feature selection, layer management, ...).

Route Storage component, which stores the locations from a particular remote vehicle in the database using the Persistence component, therefore building a route, that can then be visualized or analyzed.

Route Analysis component, which offers some basic tracking queries, which can be performed on a route, and a GUI for building more complex tracking queries.

Persistence component. All these components and applications need some persistence for their data, as they need to store the locations they receive or generate for further analysis. Usually, this persistence is provided by a database manager, such as Oracle® [10], which is used through standard mechanisms as ODBC or using special “ad-hoc” libraries, which is the case of Oracle SQL embedded sentences [11].

These components can be combined in order to build different systems. These ones should be able to solve different kind of problems such as distributed GPS data collectors, local area control centers and a headquarter control, or distributed management of the same GPS data.

Next, we are going to present two sample deployments of the system. They have been developed as a laboratory test bed and a fair presentation. Section 3 presents some use cases of the components deployment based on currently operative systems developed, or advanced designs built for some competitive tendering.

Sample deployment: Laboratory test bed.

For a further explanation of the system, **Fig. 2.** shows a sample deployment that we use in our R&D laboratory as a test bed for real systems.

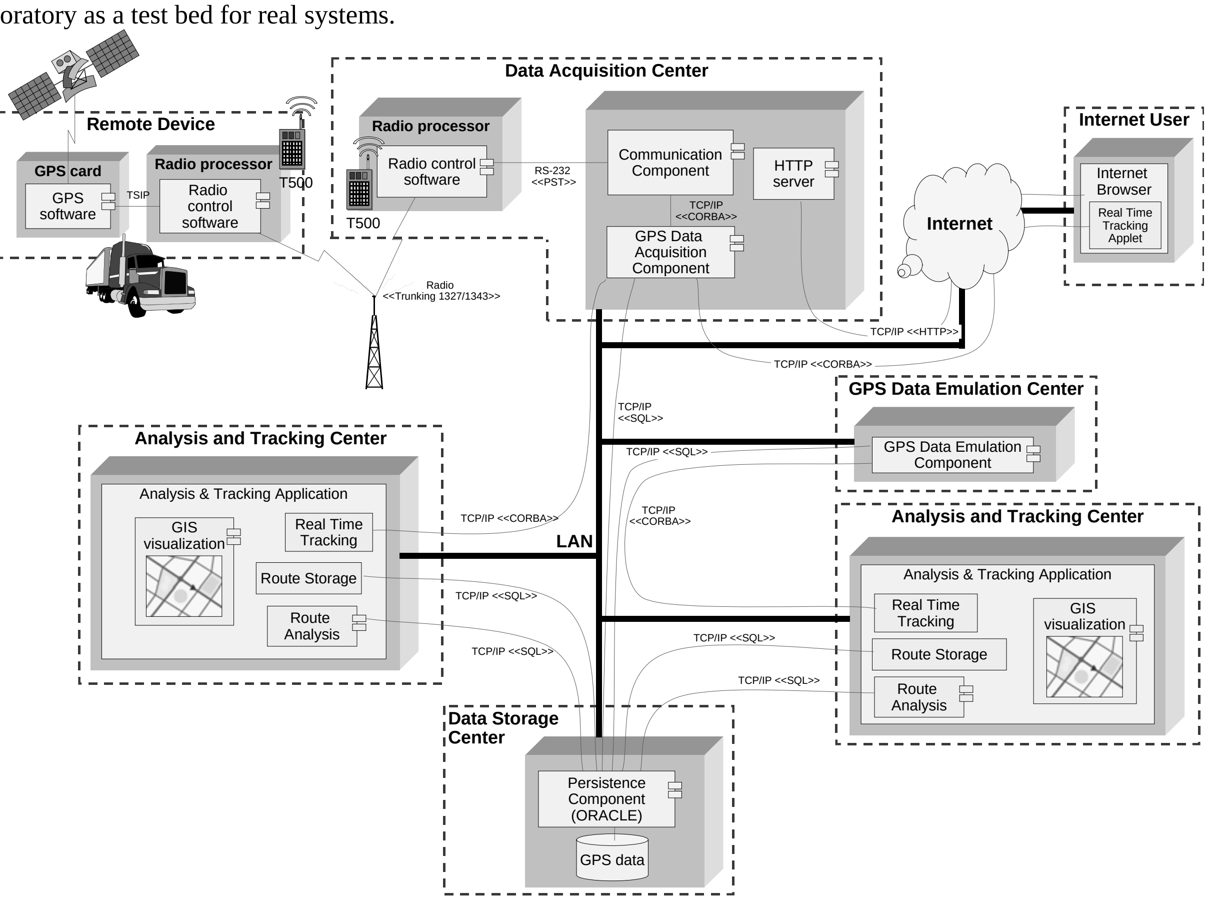


Fig. 2. Sample test bed deployment.

The remote device is a single physical equipment that consists of a trunking [15] radio with an integrated GPS card. It has simple software that lets the GPS card be commanded remotely. This device can be asked for its position at any moment, but it can also be configured to send its location automatically, each time a certain amount of time has passed (periodical sending), or each time it moves a certain distance (threshold sending).

In our laboratory we have a set of nodes connected through a LAN, each one running a certain set of components and applications.

First we have a data acquisition node, which is a computer with a radio device connected to it through a serial port. This computer runs both the Communication and the Data Acquisition component. The Communication component sends and receives messages from the radio device. It is a CORBA server, so that the radio device may be used from any other node in the LAN. The Data Acquisition component acts as a client of the Communication component. It decodes the locations received from GPS remote devices and builds any necessary message that may be sent to the remote devices. On the other hand, this component is also a CORBA server, offering the location of the remote devices to other applications. The data acquisition component also stores, using the Persistence component, all the locations it receives in a database, building a daily table for each different remote device it controls.

All the CORBA client-server interactions [12,13] in our system work according to the Subject-Observer pattern [4]. This means that the server knows its clients and is capable of sending information to them in an asynchronous way. Clients do not need to ask the server for information, as it is automatically sent to them when necessary. This way, clients can be easily built following an event-driven method.

As this is a test bed architecture, we built a special component which allowed us to test the system with no remote device available. This is the GPS Data Emulation component. This component can be substituted for the GPS Data Acquisition component as it has the same interface for its clients. The only difference is the source of GPS data. While in the Data Acquisition component, the data are real-time locations from true remote devices, the GPS Data Emulation component gets its data from historical databases. It works upon stored routes, either received from true GPS devices or artificially generated. Then, it emulates the behavior of a real GPS device that is following the stored route. It can even interpolate new locations among the given stored locations if it is required.

We have several nodes with a final application for tracking remote vehicles. These applications can work against either the Data Acquisition component or the GPS Data Emulation component. The application is built upon the GIS Visualization component and shows the received locations over digital maps of the zone. It also offers GUI (Graphical User Interface) for configuring the GPS devices in order to receive locations more or less frequently. Besides, it can store the locations received from a particular remote device, forming a route. These routes can then be analyzed using the Route Analysis component, which is also integrated in the final application.

All these components and applications store their locations and routes in an Oracle® database. Every interaction with this database is implemented through embedded SQL sentences, using special libraries provided by Oracle®.

Sample deployment: Stands in a fair

This system was shown in a professional fair that took place in Perigueux (France) in October 1998. There were two stands showing the tracking features of the system. But there was only one vehicle with a remote device to be shown in both stands (see **Fig. 3.**).

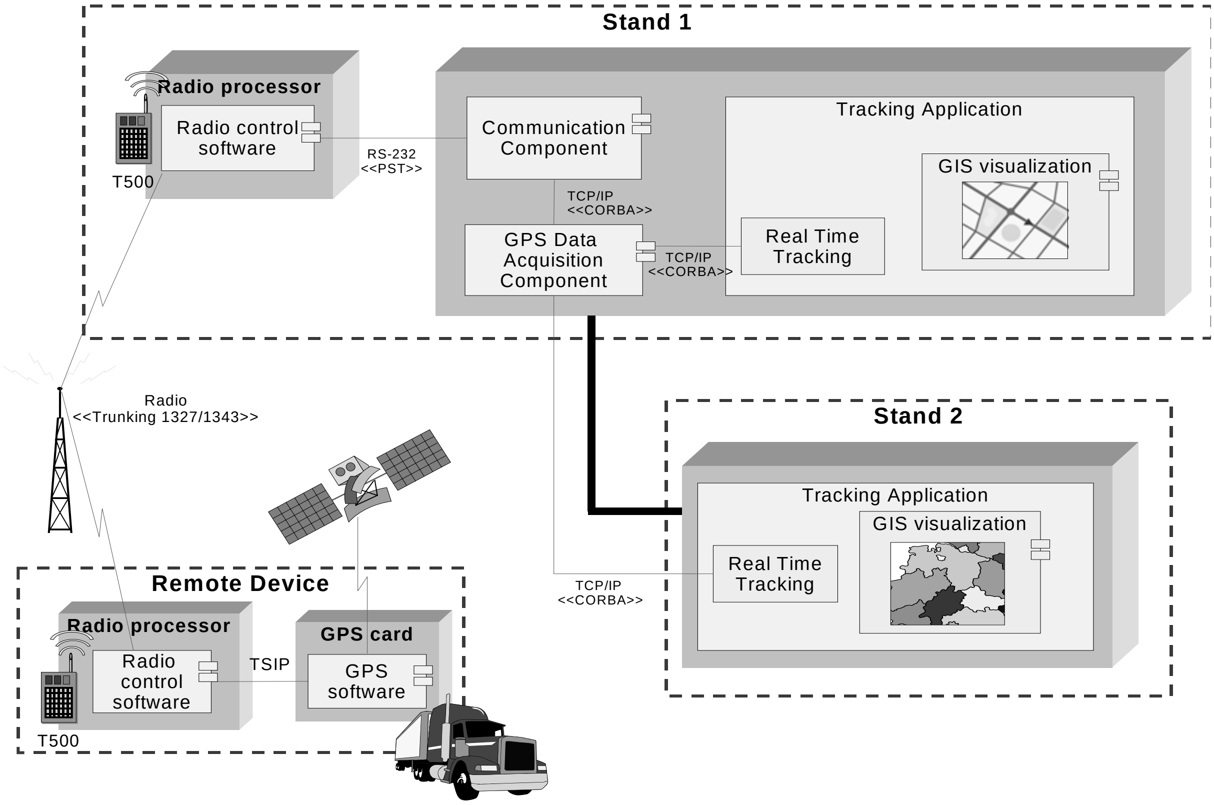


Fig. 3. Two stands showing the system.

Therefore, the stands were connected through a network and a complete system was installed in one of them, while the other one was only running a monitoring application. This application worked as a client for the system in the first stand, so the same vehicle was shown in both stands. Moreover, as the two monitoring applications were independent, the vehicle could be shown over different maps and with different visual characteristics in each stand.

Use Cases of the system

Snowploughs tracking

This system was intended to be used in France for monitoring the location of snowploughs across one Department (the political distribution in France is organized in Departments), from a single office. However, due to the orographic characteristics of some of the French Departments, there were some different zones of radio coverage without any link among them. The snowploughs could be on any of these zones and move freely from one to another at any time. It was necessary a real time tracking, but there was no possibility of getting the snowploughs' locations continuously via radio.

The best solution find is to distribute the GPS data acquisition process. In order to do that, the system is deployed as could be shown in **Fig. 4.**.

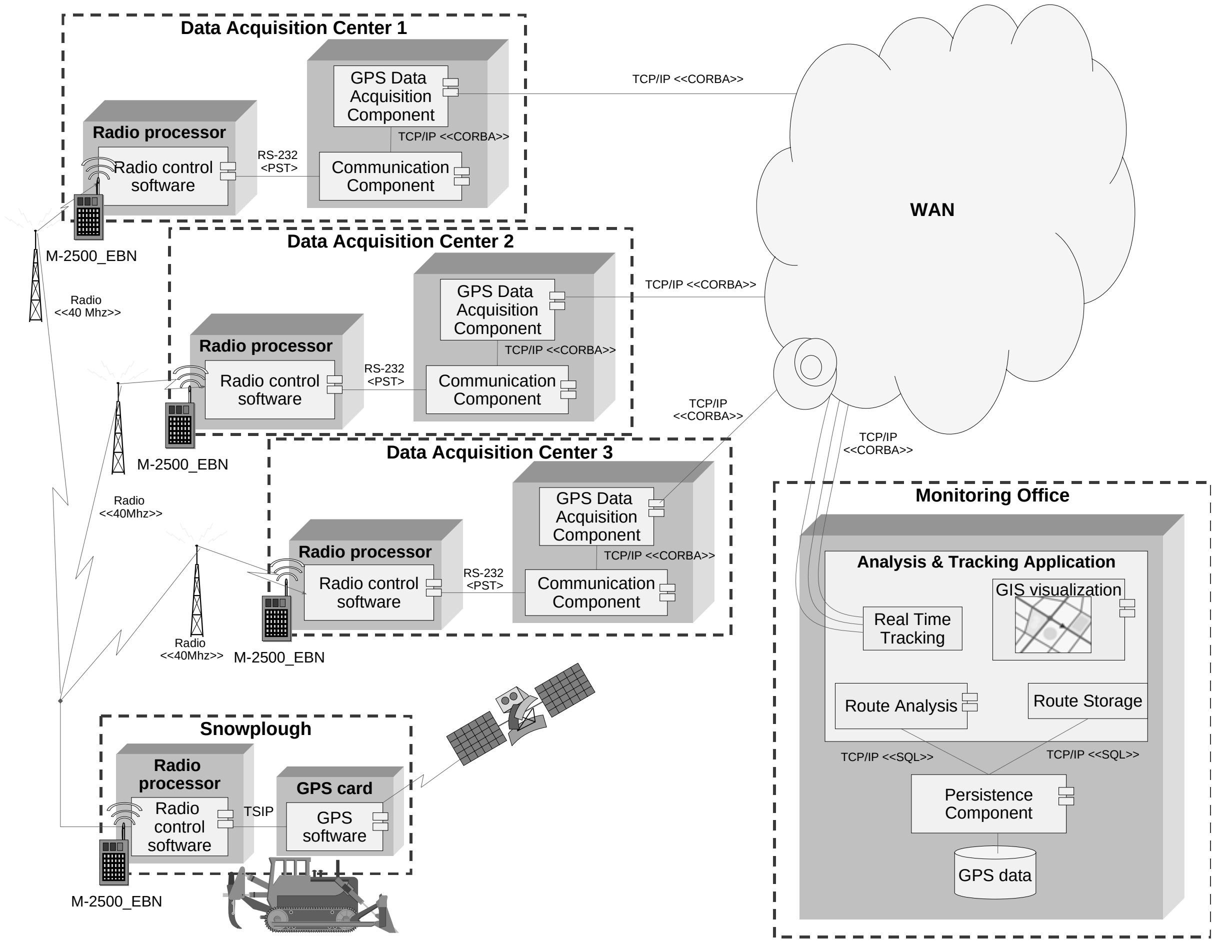


Fig. 4. System architecture for the ‘snowploughs tracking’ use case.

This configuration has a data-acquiring center on each of the coverage zones. The remote devices in the snowploughs send their locations to the receptor that is available at the moment, depending on the coverage zone where they are. The three data-acquiring centers are connected to the monitoring office through a WAN (Wide Area Network) . In the monitoring office, there is a tracking application that receives the locations of the snowploughs from the three data-acquiring centers and displays these locations using a GIS component. This application is also capable of storing and analyzing routes, by means of an Oracle® database.

Ambulances network management

When someone has to be rushed into hospital, the response time is of vital importance. One interesting solution is to use a GPS-based fleet control system, which permits to offer a fast and efficient service when a help call takes place. That is how a private ambulance company, which render service all over Aragon, a Spanish Autonomous Community[2], understood it.

The main targets are to know the exact location and the state (on-duty or off-duty) of all its ambulances at all times and to plan the ideal route to arrive at the suitable medical center from the help call place. All this to render an efficient and quickly service when someone need it. Actually, each ambulance has a radio equipment to communicate with the control center, which constantly informs them of the urgencies to answer and provides them information about the medical center where to arrive with the patient. To know the actual position of each ambulance, the operator, who works in the control center, makes a radio call to all ambulances that render service near urgency’s place. Then he decides what off-duty ambulance comes to the aid.

As a solution a GPS-based fleet control system has been planned. In each ambulance, it is installed a GPS card integrated in its radio equipment that periodically determines the exact location of the vehicle (see Figure 4). This information and the state of the ambulance are sent through the radio to the fleet control center. So, it is

[2] In Spain, the political distribution is organized in Autonomous Communities, each having a capital city. Besides, each Autonomous Community is divided in one or more Provinces, also having, each one of them, a capital city. The capital city of an Autonomous Community may coincide with the capital city of one of its Provinces, but this is not always true.

possible to know at any moment what off-duty ambulance is nearer of the help call place. This information can be represented on a digital map and displayed on a video-wall to make the operator's decisions easier. Besides, if the system has present information of the situation of the traffic along Aragon Autonomous Community roads and the hospital situation of each medical center, the system will help the operator to optimize the route and the destination that the ambulance must cover working as a Decision Support System (DSS) [3].

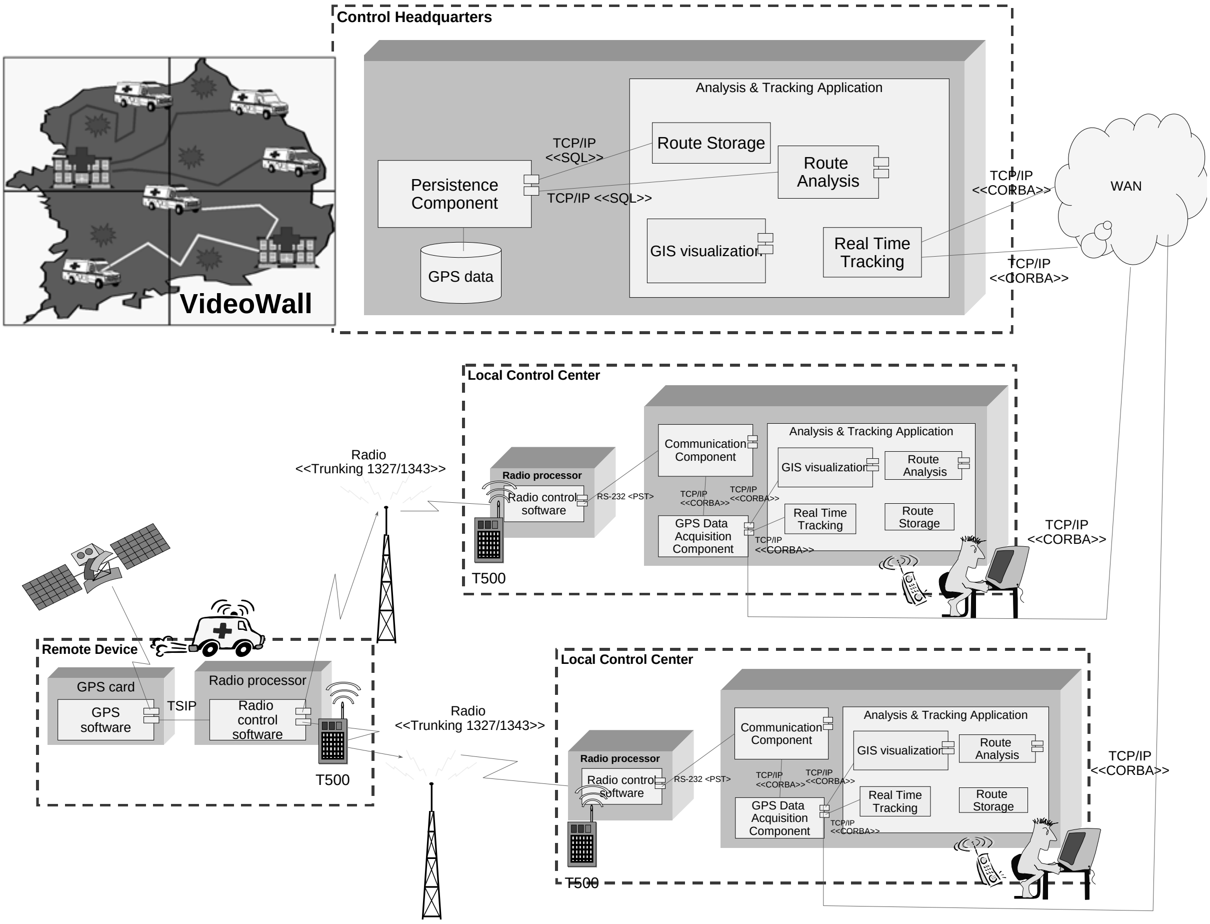


Fig. 5. *System architecture for the* 'ambulances network management' *use case.*

As it could be shown in **Fig. 5.**, the system is structured in two types of control entities: Control Headquarter (CH) and Local Control Center (LCC). There is one CH, which is responsible for keeping a global, real and present vision of the information of the system (location and state of each ambulance, information and state about the traffic and the medical centers,...), and some LCCs, where the help calls of the respective covered area are received and it is decided what available ambulance realizes the urgency service. Each LCC reports about the tracking events and the local decisions to the CH, which centralizes the global situation of the fleet and stores it in a data base for its later treatment. The Aragon configuration of the system has the CM and one LCC in Zaragoza city (capital city of the Aragon Autonomous Community and one of the three Aragon Provinces), and two more LCCs, one in Teruel city and other in Huesca city (both are capital cities of the two other Aragon Provinces).

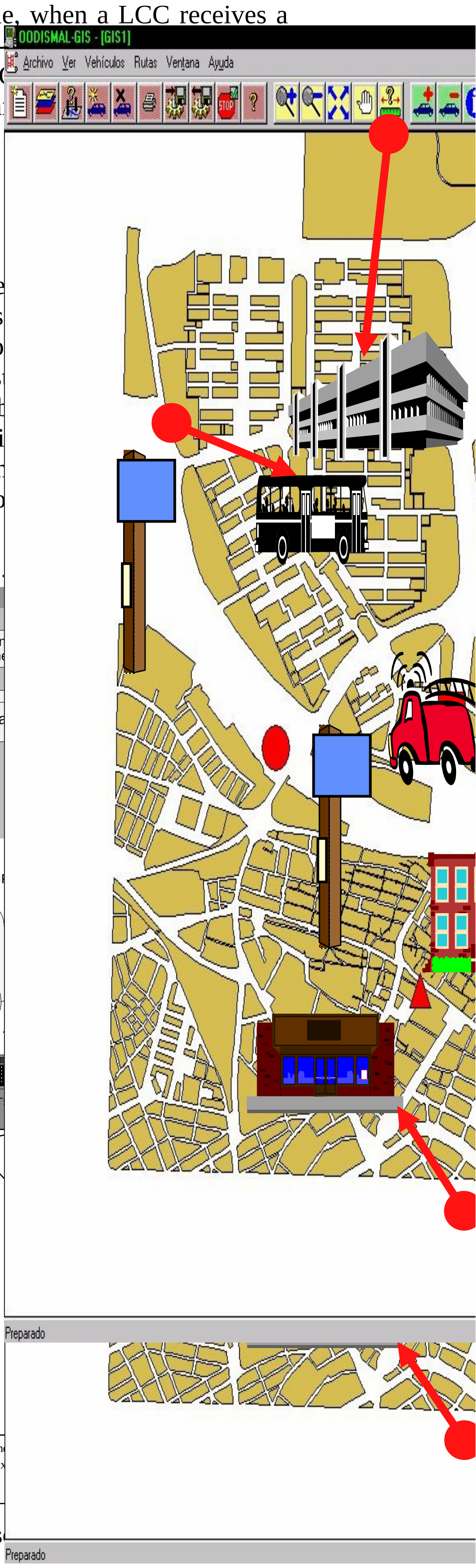
From the functional point of view, a LCC is simultaneously a data acquisition center and an analysis and tracking center. It receives through its radio processor the GPS locations and the state of the different ambulances. These geographical locations are the basic information to develop real time tracking and route planning, analysis and storage. Finally, to help the operator to make correct decisions this last information is represented on a GIS visualization module.

On the other hand, a CH is a global tracking center. Its data sources are the LCCs GPS data acquisitors. These LCCs send it information about the local adopted decisions.

In order to give the system an extra value, it has been designed in an open and flexible way which enables in the future the incorporation of new and similar public or private services such as the local police, the fire department or civil protection.

This system also permits to connect it through Internet to display on a home computer the location, the state and the route which will cover each ambulance of the private service. In this way, a destination medical center can know the remaining time for an on-duty ambulance to arrive and if it's even possible to receive a previous medical diagnostic.

Finally, it describes a found problem called 'overlapped local areas'. For example, when a LCC receives a help call, it is possible that its nearer off-dutty ambulance is further than an off- different LCC regarding the help call place. When this situation occurs, the first LCC call, informs the CH to make a global decision. This asks the others LCCs involved a dutty ambulance.



Monitoring and user information system for urban buses

In our days, some great cities have improved their public transport service impleme location system, for example, Paris [1]. Some targets are to have better-informed pass interval and to increase the security of the service. All it, with the intention that the p satisfied and prefer the public transport to other means. These already implanted sys expensive communications systems to solve some problems like precision troubles, b great number of bus lines. That turns them into nonviable projects to be implanted i than two hundred thousand inhabitants). In Spain, there are more than thirty cities wh 100.000 and 200.000 inhabitants. This fact motivated the design of a complete mo adjusts to the requirements of this type of cities.

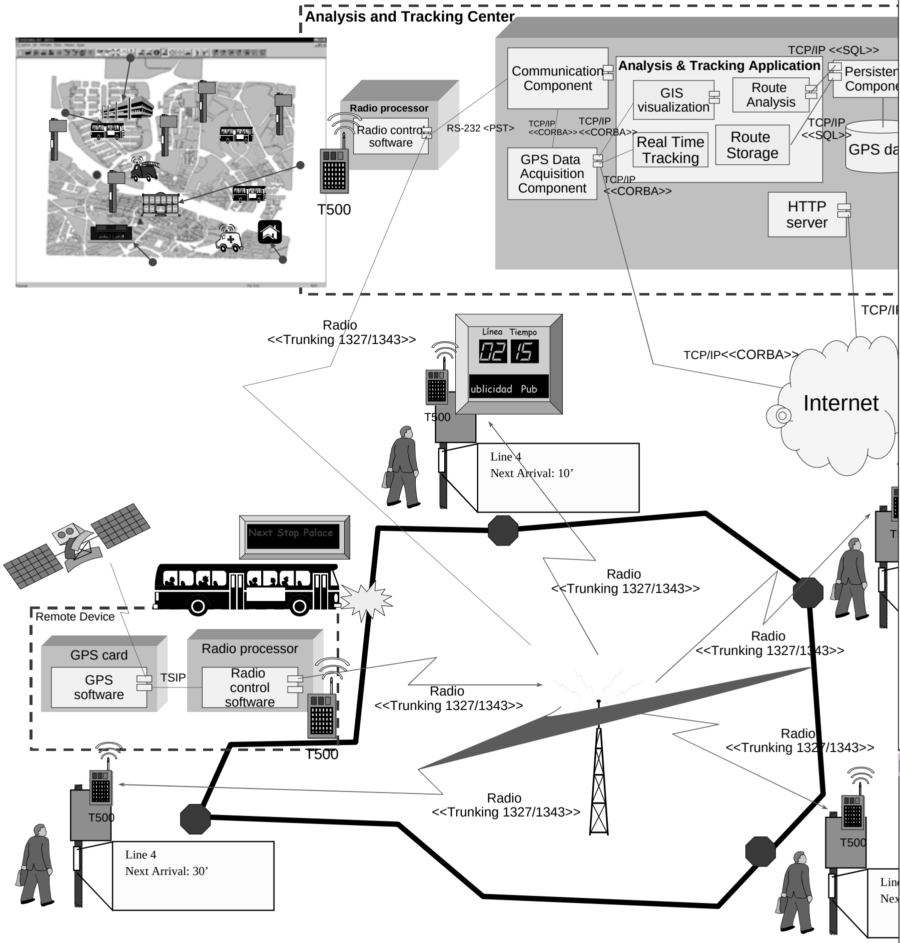
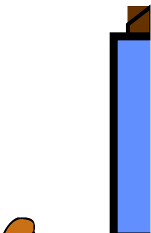


Fig. 6. System architecture for the 'monitoring and user information system for urban buses' us

The main target is to inform the passengers, who are waiting at a stop, how long they must still wait for the next bus of each line that passes that bus stop and arrives at it. It's also interesting to report users that are travelling inside each bus about the name of each stop as the vehicle arrives and display this information on a LED (Light Emitting Diode) display.

Fig. 6. shows the system architecture. The remote device embarked in each bus, the bus stop equipment and the analysis and tracking center integrates it.



The bus equipment consists of a radio device with incorporated GPS card to determine and communicate the exact location of the bus to the control center and the bus stops. In the inside-bus there is a small LED display to inform about the name of each stop, the present time or to visualize some advertisements. The radio has a voice channel so that the driver can make emergency calls from the bus.

Besides, each bus stop has a radio equipment installed to receive the information that a bus emits in a periodic way and the sent information from the control and tracking center of the company. It is also installed a microprocessor with a small memory to store the exact position of each stop and the estimated time that takes to cover the distance between two consecutive stops of each one of the lines of the stop. To visualize bus intervals, there is a display situated at the top of the bus stop.

Each bus detects when it is going to arrive at a stop and emits the GPS location where that stop is located, making a group radio call (a call made to a set of radios, instead of a single one) to the rest of stops of the bus line and the control center. Each stop keeps inside the memory of the microprocessor stored the estimations of intervals between consecutive stops and their locations. Known this information and the actual bus location, the microprocessor of each stop can calculate the time that takes the bus to arrive at it. Once the result is available it is visualized in the bus stop display.

Other additional devices can be used to improve the detection of stops that could replace or complement to the reception of locations GPS emitted from the buses (sensors to detect the opening of doors of the bus, detection of stopped vehicle, some button pressed by the conductor indicating the arrival to a stop...). This aim is to avoid some weakness in the intervals visualized in the bus stop’s display. For example, it must not appear that the bus will arrive in two minutes when the bus is already held up in the stop.

It is possible that between two consecutive stops the real time to cover the distance among them is higher than the estimation, being delayed to the arrival of the bus to the next stop. This can happen for example, by dense traffic. In these exceptional cases, when the bus arrives at the following stop, the system synchronizes, calculating new valid estimations.

The urban bus company has a control center where it is received all the location information emitted from the buses of the fleet. From this control center the starting and the maintenance of the system are controlled. In a future, it will be possible to integrate the functionality of an operation aid system. The system makes fleet control and tracking tasks, visualizing on digital maps the exact position of each bus and the time that takes it to arrive to each stop of the line that it covers. The system also permits to adjust schedules as necessary to eliminate strings of buses and respect regular intervals at the stops. The information of tracking, as well as the events or alarms happened during the route of each bus are stored to be analyzed later. This analysis serves to improve the interval planning between consecutive stops and to identify problems that are regularly repeated, reacting to them earlier.

A user can connect itself from his house through Internet to the web site of the urban bus company to consult when it is estimated that the buses that render in a line are going to pass by one of its stops. The user will also be able to see on a digitized map of the city the route defined for each line of buses and in what points of that route are the respective buses that operate in that line. As the tracking center of the company receives new information on the locations of the buses, the tracking information of the fleet offered through the Web will be updated so that the users have a real time vision at any moment. Similarly, in order to offer precise information to the users taking advantage of the previous functionality, information places can be installed in different points of the city.

Conclusions

This work has presented a general idea of the possibilities that the use of distributed real-time GPS data collector can offer in the development of vehicle location systems. This has been done by presenting three systems that have an architecture which includes distributed real-time GPS data collectors.

These systems are based in a library of distributed components to integrate radio communications, real-time data captured by GPS units, on-line and off-line position analysis, and GIS components we have developed. Therefore, we can distribute several real-time GPS data collector achieving some valuable advantages. It is possible to adapt the system to environments with radio coverage problems (as shown in the snowploughs use case), as well as balancing communication load among several real-time GPS data collectors, in the case of communication over-headed systems (as the ambulances use case). Besides, having several collectors provides higher reliability and robustness for the system against network problems and collector working failures.

As it is shown in the urban buses use case, another valid alternative to distributed real-time GPS data collectors is to take advantage of the communication network to distribute real-time GPS data. This permits the installation a lot of systems (information places, monitoring centers,...) which are able to exploit real-time GPS data over a large zone, avoiding the cost and the slowness of wide area computer networks.

References

1. A. Ampélas, M. Daguerregaray. “Paris Public Transit The GPS Difference”. *GPS World* October, 1999. pp 36-40
2. A. Boulmakoul, R. Laurini, S. Servigne, M.A. Janati Idrissi. “First specifications of a telegeomonitoring system for the transportation of hazardous materials”. *Procd. Of the TeleGeo 1.999*. pp 47-52.
3. V. Dhar, R. Stein. *Intelligent Decision Support Methods. The Science of KnowledgeWork* Prentice Hall International Editions. 1997.
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides. *Desing Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley Publishing Company. 1994.
5. J.L. Infiesta, J. Roca, J.L. Pérez. “¡Todos a bordo! Sobre el carril con los trenes de Cataluña”. *GeoConvergencia*, may 1998. pp 20-29.
6. E.D. Kaplan (ed.). *Understanding GPS Principles and Applications*. Artech House Publishers. 1996.
7. J.L. Long. “Black Thunder’s Roar: Mining for Solutions with RTK GPS”. *GPS World*, march 1998. pp 22-28.
8. P. Muro-Medrano, D. Infante, J. Guillo, J. Zarazaga, J. Bañares, "A CORBA Infraestructure to Provide Distributed GPS Data in Real Time to GIS Applications." *In Computers, Environment and Urban Systems*, v. 23, p. 271-285, 1999
9. Object Management Group. Object Management Architecture Guide, Version 3.0. Framingham, MA: Object Management Group, 1995.
10. Oracle8. Release 8.0.4. *Getting Started for Windows NT.* Oracle Corporation, 1997.
11. Oracle. Release 2.1. *Programmer’s Guide to the Oracle Pro*C/C++. Precompiler.* Oracle Corporation, 1994-1995.
12. R. Orfali, D. Harkey and J. Edwards. *The Essential Client/Server Survival Guide CORBA.* Wiley Computer Publishing. 1997.
13. R. Orfali, D. Harkey. *Client/Server Programming with Java and CORBA.* Wiley Computer Publishing. 1998.
14. C. Szyperski. *Component Software. Beyond Object-Oriented Programming*. Addisson Wesley, 1997.
15. F.J. Zarazaga, J. Valiño, S. Comella, J. Nogueras, P. Muro-Medrano: “TRUNIS: an Object Oriented Trunking Radio Telephone Network Information System. An experience report”. *Proceedings of the 29 Conference on Technology of Object-Oriented Languages and Systems. IEEE Computer Society Press* pp. 251 - 260. Nancy, Francia. Junio 1999