WebServices Integration on an RFID-Based Tracking System for Urban Transportation Monitoring*

Jose I. San Jose†, Juan J. de Dios, Roberto Zangroniz, Jose M. Pastor
Institute of Audiovisual Technologies (http://itav.uclm.es)
University of Castilla-La Mancha
Cuenca, Spain
{JoseIvan.SanJose, JuanJose.deDios, Roberto.Zangroniz, JoseManuel.Pastor}@uclm.es

Abstract

Automatic Vehicle Location (AVL) Systems are being introduced increasingly in many cities around the world. These systems are aimed at cost reduction and optimization of time and resources. A Radio Frequency Identification (RFID) based system called RFID Urban Transportation Monitoring System (RUTMS), is proposed in this paper for urban transportation tracking. The monitoring of bus stops and lines on the route map has been integrated and an EPCGlobal-based WebServices interface to the proposed system has been developed. RFID readers are located in the bus stops for retrieving real-time data, as these data can be compared to the information of the transport routes stored in the database of the tracking system. A Web environment is proposed for this purpose, as these data can be accessed anytime through whichever device using a web browser. The architectural model is based on a Fosstrak EPCIS Repository, as it has been certified by EPCGlobal, and it can be used in any open source software platform implementing the EPC Network specifications. Two Web services are developed: the first one is used for monitoring the transport location and the second one is used for displaying the arrival time of any bus to a specific bus stop. A graphical interface has also been integrated on the tracking system. The system we propose is a simple and inexpensive tool to locate bus transport in a Geographic Information System (GIS), by using RFID readers in the bus stops and RFID tags in the buses.

Keywords: EPCGlobal, Web Services, RFID, location, AVL.

1 Introduction

Automatic Vehicle Location (AVL) [5] provides real-time location information for any mobile assets upon which it is installed. AVL is used for different traceability purposes, especially for those related to tracking one vehicle or a fleet of vehicles. Tracking system technology [2] was made possible by the integration of several navigational technologies, as Global Positioning System (GPS) [22], Geographic Information System (GIS) [7] and General Packet Radio Service (GPRS) [13]. Also, AVL can be used together with other technologies such as Radio Frequency Identification (RFID) [35] or Wireless Sensor Networks (WSN) [4].

IT CoNvergence PRActice (INPRA), volume: 1, number: 4, pp. 1-23

*This paper is an extended version of the work originally presented at the International Workshop on Pervasive Internet of Things and Smart Cities (PITSaC-2013) in conjunction with AINA-2013 International IEEE Conference, Barcelona, Spain, March 2013, titled "RFID Tracking for urban transportation using EPCGlobal-based WebServices"[27]. A more detailed description of our proyect is added. Besides, the Web interface is described in greater detail. Also made a comparison with other projects, that use GPS or RFID to locate public transport on a map.

†Corresponding author: Institute of Audiovisual Technologies, University of Castilla-La Mancha, Cuenca (Spain). Tel: +34 969-179-100 ext. 4773, email: JoseIvan.SanJose@uclm.es

RFID technology [20] is one of the most innovative technologies in object location and identification, as it provides a good solution for the traceability and tracking of any kind of products in a simple and economic way. On one hand, the application of RFID technology can benefit consumers through improved product availability, speed of service, and quality assurance. RFID tagging of products by manufacturers, wholesalers, and retailers appears to be the most promising approach to reliable product tracking and tracing, instead of bar codes.

On the other hand, RFID helps companies to improve the supply chain efficiency, and provides a high added value by optimizing the distribution process and the available resources [19]. Besides, tags can be used for identification, inventory, and product traceability.

The Electronic Product Code (EPC) [15] [17] is designed as a universal identifier that provides a unique identity for physical objects. EPC is generally thought of as the next generation of the standard bar code. It is used in information systems that need to track or otherwise refer to physical objects. A very large subset of applications that use the Electronic Product Code also rely upon RFID Tags as a data carrier, and enables the assignation of a unique global number to any product, so that it can be associated with specific product information, such as date of manufacturing, origin and destination of the shipment.

In this paper, we design a storage system for the events detected in the RFID readers located along the distribution chain of the products, from their manufacturing till they are delivered to customers. Data are stored and retrieved using an RFID tag which transmits the ID of the product when prompted by the RFID reader.

Our main goal is to design and implement an EPCGlobal RFID-based system, to store tracking and tracing data events. These stored events may be accessed via a Web service specifically devised for this purpose, with the capability to be accessible from any device, either a PC or any mobile device (smartphones, tablets, etc.).

We propose a system, called RFID Urban Transportation Monitoring System (RUTMS), that will ensure a set of minimum requirements of scalability, extendibility, and compatibility. In order to meet these criteria, EPCGlobal standard is fulfilled. In addition, the proposed system also provides the necessary tools to building our system. A simulation of the system will be implemented, as it is not possible to achieve all the resources required for this implementation at the moment.

Another goal of this paper is to locate any element through a GIS, Google Maps JavaScript API v3 [33] [11] is used. Google Maps is commonly accepted, and globally referenced due to its usability.

The architecture of RUTMS is based on the Fosstrak EPCIS Repository, as it is a complete implementation of the EPCIS standard specification, including what we need to deploy the application. Besides, it is also certified by EPCglobal. Futhermore, in this paper we also perform a comparative study with other similar systems using RFID or GPS to locate vehicles.

In this paper, we describe the general methodology that could be defined to develop any other type of similar system. Finally, and as a particular case, we will implement an application for monitoring the transport location and the arrival time of a bus route. This application is defined as a Web Service. The user queries the information of the current situation of a bus, the system will display it on the map and the time left to reach to the selected bus stop will be also indicated on the route.

2 Related Works

At present, beyond the simple management of vehicles [6], which comprise a fleet, different ways of working are essential, related not only to the management of the vehicle itself, but going a step further, related to the goods carried within them. Therefore a new set of technologies is necessary to meet the needs that allow us to be able to offer services that enable monitoring, tracking and traceability of products contained inside the cargo area of vehicles that come into play to implement this new solution.

AVL Systems are being introduced increasingly in many cities around the world [26]. The objective is to improve the efficiency of the road-based passenger transport systems. Satellite-based location and communication systems, particularly the GPS, have been the infraestructure needed for AVL systems, which are now supporting several types of applications as: Real-Time Passenger Information (RTPI), Fleet Management and Operations (FMOs) and Public Tansport Priorities (PTPs).

On one hand, most of the AVL Systems are using GPS as the main location technology. London's iBus [9], Sydney PTIPS [28], Auckland SR/RTPIS [34] or Glasgow BiAS [10] are examples of public transport that use GPS as location technology.

On the other hand, we can find AVL Systems based on RFID. There are systems that use RFID and WSN [25] or automatic vehicle tracking [3].

In a more visual way, we compared the different systems in two tables. Table 1 shows a comparative between RUTMS, and the systems mentioned before. In this table, we compare the location technology, data transmission technology, data storage location, data processing location and bus priority at traffic signals of each system.

Table 2 shows the different applications that have been developed in each system, as real time / estimation location, passenger information display, app for mobile devices, route calculation and number of buses arriving to a bus stop.

For the tracking tasks described above, GPS technology has already been well established, by which it is possible to determine the absolute position of a particular vehicle, whose position can be associated with an existing map, obtaining additional information about the path that the vehicle on which the receiver is installed.

Also, RFID technology allows to determine the position of a particular vehicle, when the vehicle passes near an RFID Reader. The measurement distance depends on the kind of tag attached to the vehicle (active or passive), also depends on the position of the tag on the bus, and the RFID antennas.

3 Technologies

For developing our system, the following technologies are used: RFID, Electronic Product Code Information Services (EPCIS)[14], Fosstrak EPCIS Project [29] and Web Services [23].

3.1 RFID

Radio frequency identification (RFID) is a generic term that is used to describe a system that transmits the identity (unique serial number) of objects or people wirelessly, using radio waves. It's grouped under

	Location Technology	Data transmission technology	Data storage location	Data processing location	Bus priority at traffic signals
London (iBus)	GPS	GPRS / WLAN	AVL Centre and Bus	AVL Centre (lateness) and Bus	Yes
Sidney (PTIPS)	GPS	GPRS	AVL Centre and Bus	AVL Centre	Yes
Auckland (RTPIS)	GPS	GPRS	AVL Centre and Bus	AVL Centre	Yes
Glasgow (BiAS)	GPS	Radio / LAN	AVL Centre and Bus	Bus (lateness)	Yes
BMS using RFID and WSN	RFID	WSN	AVL Centre	AVL Centre	No
RUTMS	RFID	LAN / WLAN / WSN	AVL Centre	AVL Centre	No

Table 1: Comparative of technology

	Real time / estimation location	Passenger information display	APP for mobile devices	Route calculation	Number of buses will arrive to bus stop
London (iBus)	Real time	Yes	No	No	1
Sidney (PTIPS)	Real time	Yes	No	No	1
Auckland (RTPIS)	Real time	Yes	No	No	1
Glasgow (BiAS)	Real time	Yes	No	No	2
BMS using RFID and WSN	Estimation	Yes	No	No	1
RUTMS	Estimation	Yes	Yes	Yes	2

Table 2: Comparative of applications

the broad category of automatic identification (Auto-ID) technologies, which have been used to reduce the amount of time and labor needed to input data manually and to improve data accuracy.

Bar code systems often require a person to manually scan a label to capture the data. RFID is designed to enable readers to capture data on tags and transmit it to a computer system, without needing a person to be involved.

A typical RFID tag consists of a microchip attached to a radio antenna mounted on a substrate, and

contains the identification data of the object to which are attached. The RFID tags may generate a radio frequency signal related to the data, so it can be received by an RFID reader, which retrieves the data stored on the tag and converts it to a digital format according to the specific application.

Currently, the organization of companies and universities around the world, called EPCGlobal (formerly Auto-ID Labs), is responsible for developing standards and the necessary technology to be able to identify any object effectively. In 2003, the RFID technology really began to spread through the industry, driven by technological advances, the reduction of costs and the efforts of the Department of Defense of the United States.

3.2 Electronic Product Code Information Services (EPCIS)

EPCIS is a standard designed by EPCglobal to allow sharing and pooling of data between different companies. The aim is to allow the participants of the EPCGlobal Network get a common view of the disposition of the various objects within the business environment.

The Architecture Framework published by EPCglobal [16] provides a comprehensive overview of the EPCglobal standards. It shows how the different interface standards are related and outlines the principles that have guided the design of the standards.

EPCIS defines standard interfaces that allow the EPC data being captured and later be used by operations and an associated data model. Capture and query of these data, includes the use of persistent databases and the sharing of information application-to-application.

The EPCGlobal Architecture Framework[1] (see Figure 1) is a collection of interrelated standards for hardware, software, and data interfaces, together with core services that are operated by EPCglobal and its delegates, all in service of a common goal of enhancing the supply chain through the use of Electronic Product Codes.

This framework does not dictate particular system architecture, but leaves this to implementers or manufacturers who can choose the system architectures that are most appropriate for their deployments. EPCIS only specifies the interface between applications and data collection needed to access them.

It does not specify how to implement the operations or databases, but in no case the definition of the interfaces will be outside of the framework scope. In our development we use the RFID reader of SkyeModuleM9 to capture data. This hardware meets the EPCGlobal standards and specifications, but it has its own communication protocol and other special features.

Therefore, EPCIS is a standard that has a layered structure, extensible and modular. This feature offers a number of advantages such as, for example, the reuse of code. The different layers described within the EPCIS standard: Abstract Data Model Layer, Data Definition Layer, Service Layer and Bindings are presented in Figure 2).

These layers are described briefly below[2]:

[1]More information about data standards, interface standards and standards in development at: `http://www.gs1.org/gsmp/kc/epcglobal`

[2]More information about the layers at: `http://www.gs1.org/gsmp/kc/epcglobal/epcis/epcis_1_0_1-standard-20070921.pdf`

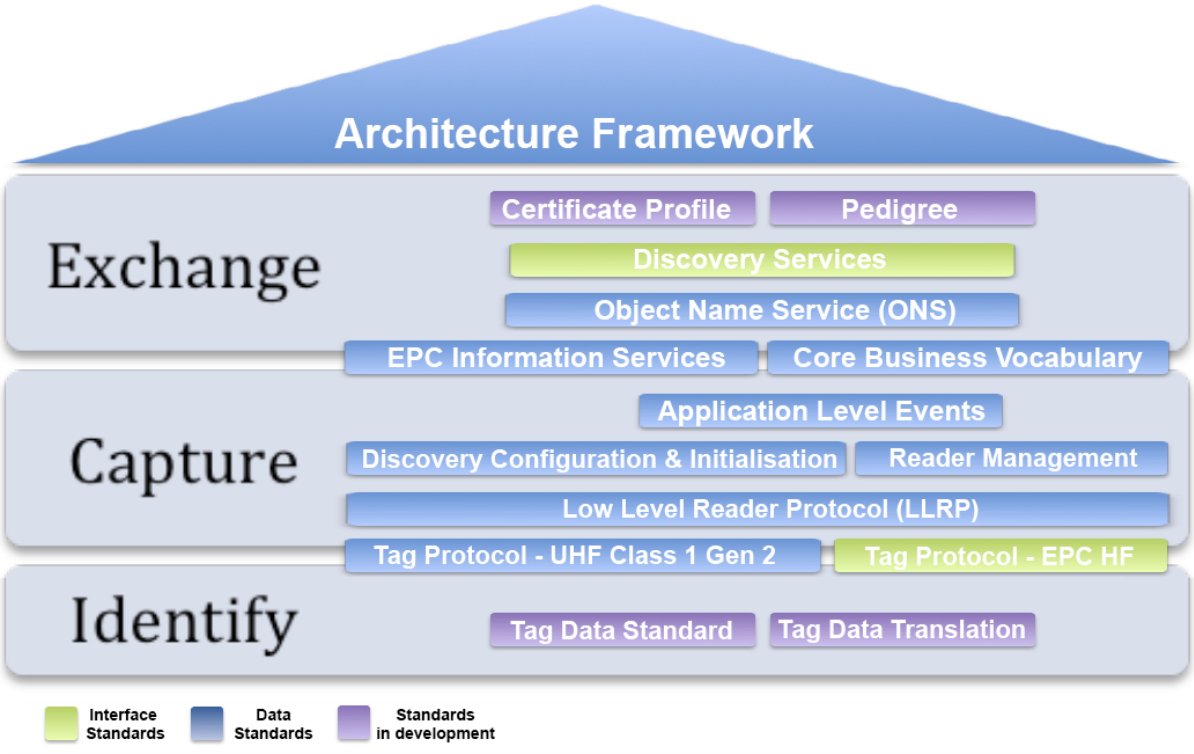

Figure 1: EPCGlobal Arqhitecture Framework

- The Abstract Data Model Layer specifies the generic structure of EPCIS data and this layer specifies the general requirements for creating data definitions within the Data Definition Layer.

- The Data Definition Layer specifies what data is exchanged through EPCIS, what its abstract structure is, and what it means. Data definitions in the Data Definition Layer are specified abstractly, following rules defined by the Abstract Data Model Layer.

- The Service Layer defines service interfaces through which EPCIS clients interact. Two Service Layers are defined: the Core Capture Operations Module and the Core Query Operations Module.

- Bindings specify concrete realizations of the Data Definition Layer and the Service Layer. In this specification, nine bindings are defined for the modules defined in the Data Definition and Service Layers.

The layered technique for specification promotes extensibility, as one layer may be reused by more than one implementation in another layer. Besides the inherit extensibility, EPCIS provides us two specific methods to do the extensibility: Subclassing and Extension Points.

- Subclassing. A subclass is a new type that includes all of the fields of an existing type, extending it with new fields. An instance of a subclass may be used in any context in which an instance of the parent class is expected.

- Extension Points. Data definitions and service specifications also include extension points that may use to provide extended functionality without creating subclasses.

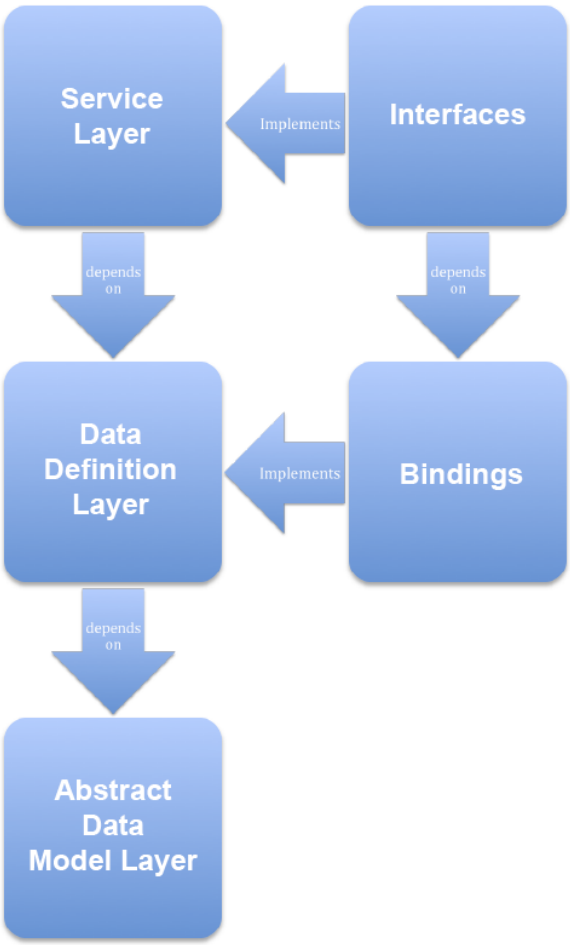

Figure 2: EPCIS Layers

The modularity consists of several single associated specifications. This property allows to EPCIS grow and envolve in a distributed way. Together with the layered structure and extensibility, the modularity allows the definition of the structure in different documents as well as promote the consistency of the structure.

3.3 Fosstrak EPCIS Project

Fosstrak EPCIS Project is an open source software platform that implements the GS1 EPC Network specifications. It provides an EPCglobal-certified EPCIS Repository as well as Query and Capture clients.

Fosstrak EPCIS Project features are, as follows:

- Deploying an EPCIS Repository.
- Run queries to an EPCIS Repository through a Graphic User Interface (GUI).
- Adding information to an existing EPCIS Repository through a GUI. This GUI is different than the before GUI.

In addition to these standards-compliant modules, it also offers a ”Web adapter” for easy EPCIS access via web protocols (e.g., REST). Fosstrak has a regular client-server architecture consisting of three modules: the Repository is the server, Query and Capture clients.

The repository is responsible for analyzing the requests of the clients, which are processed according to the rules defined in the specification of the standard. It implements five bindings as defined by the EPCIS specification: XML (eXtended Markup Language) for Data Definition Layer, HTTP for the Capture Interface, SOAP/HTTP for the Query Control Interface, HTTP and HTTPS for the Query Callback Interface.

The clients communicate with the repository via transport protocols. These protocols are: XML over HTTP to Capture client and SOAP (Simple Object Access Protocol) over HTTP to Query client, as we can see in Figure 3.

To capture EPC data, it is necessary to wrap the data into XML conformin the EPCIS XML Schema for event types. This EPCIS event must then be sent in an HTTP POST request to the repository's capture interface. An example of capture event, serialized into XML, is shown in Figure 4.

Queries for EPCIS Events must be inside a valid SOAP request to the repository. An examle of XML is shown in Figure 5.

Both clients, Capture and Query, have a development client API and a client GUI. The API is used by other applications to send queries to the repository. Manually, the interface is used to send events within the repository. Fosstrak also provides a set of libraries and documentation to be able to integrate clients Capture and Query in an architecture type J2EE.

3.4 Web Services (WS)

Web Services are technologies that integrate a set of standards and protocols to exchange data between applications developed in different programming languages and they can run on any platform. We can use the Web Services to exchange data in both private computer networks and the Internet.

Interoperability is achieved by open standards. Organizations such as OASIS and W3C are responsible for indicating the type of architecture and Web services regulation. Web Services are loosely coupled software components that offer standardized interfaces based on mainly two languages: the Web Service Definition Language (WSDL) which is used to define the syntax of the interfaces, and SOAP which defines the format of messages that are exchanged when invoking services.

In the future Internet [24] , real-world devices will be able to offer their functionality via SOAP-based Web Services (WS-*) or RESTful APIs, enabling other components to interact with them dynamically. The functionality offered by these devices is often referred to as real-world services because they are provided by embedded systems that are related directly to the physical world.

Unlike traditional enterprise services and applications, which are mainly virtual entities, real-world services provide real-time data about the physical world. Armed with this additional knowledge, one can support a more efficient decision making process. Hence, the devices are providing their functionality as a Web Services can be used by other entities such as enterprise applications or even other devices. No

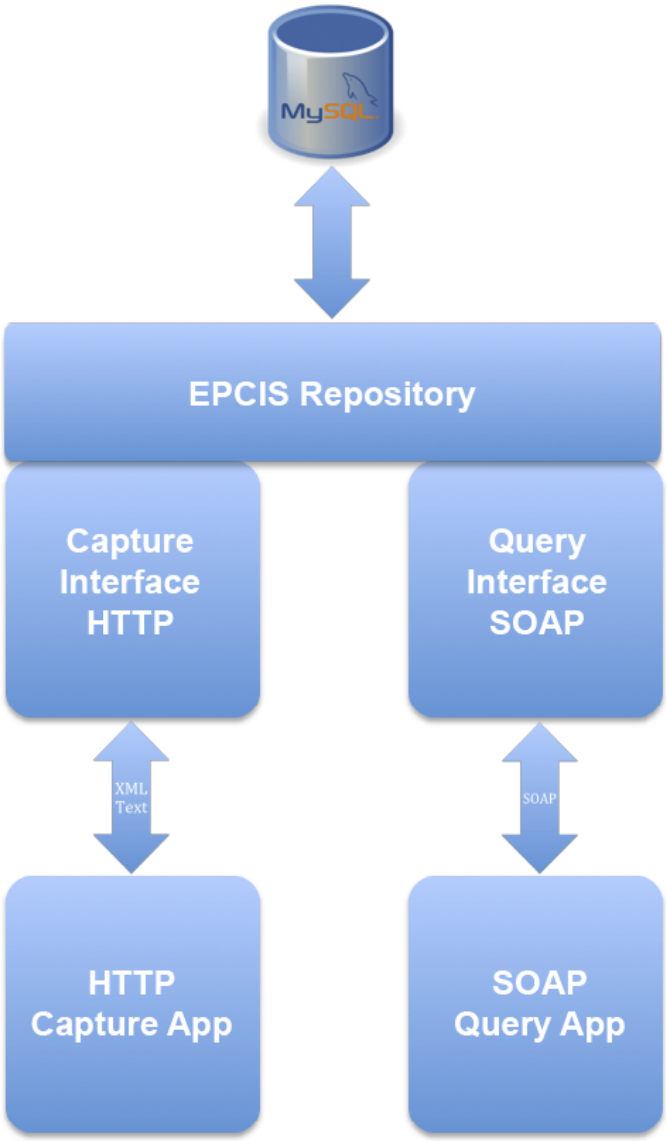

Figure 3: Fosstrak EPCIS Project Implementation

device drivers are needed anymore and a new level of efficiency can be achieved as web service clients can be generated dynamically at runtime.

4 Development Methodology

This section describes the methodology that has been proposed for the development of a generic application and that, subsequently, will serve us for the development of the application proposed in the introduction section. Our methodology is divided into three parts: design, modeling, and implementation.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<epcis:EPCISDocument
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:epcis="urn:epcglobal:epcis:xsd:1"
    xmlns:epcglobal="urn:epcglobal:xsd:1"
    xsi:schemaLocation="urn:epcglobal:epcis:xsd:1 EPCglobal-epcis-1_0.xsd"
    creationDate="2013-03-16T09:15:16.397+01:00"
    schemaVersion="1.0">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2013-03-16T09:15:16.397+01:00</eventTime>
        <eventTimeZoneOffset>+01:00</eventTimeZoneOffset>
        <epcList>
          <epc>urn:epc:id:sgtin:0614141.107346.2017</epc>
        </epcList>
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:epcis:bizstep:fmcg:shipped</bizStep>
        <disposition>urn:epcglobal:epcis:disp:fmcg:unknown</disposition>
        <readPoint>
          <id>urn:epc:id:sgln:0614141.07346.1234</id>
        </readPoint>
        <bizLocation>
          <id>urn:epcglobal:fmcg:loc:0614141073467.A23-49</id>
        </bizLocation>
        <bizTransactionList>
          <bizTransaction type="urn:epcglobal:fmcg:btt:po">
            http://itav.uclm.es/publictransport/
          </bizTransaction>
        </bizTransactionList>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```


Figure 4: XML Capture EPCIS Events

4.1 Design

Before programming, we must consider a series of requirements previously, since it is necessary to be clear about the application you want to perform.

As indicated earlier, our application must be able to show us the information that it stores in a database. To do so, we must build a Web site on which we are able to access from any device (PC, Tablet, Smartphone, etc.), whichever browser, anywhere in the world to query of the information.

To do this, our application must meet a series of functional and non-functional requirements:

1. The functional requirements are the following: RFID readers must be able to access the central database through either a microprocessor or DSP. Each hardware element reading will connect through a port serial or USB to its processor, which will also have Internet connection. The information captured by the RFID readers in the bus stops, can be sent by different ways: via ethernet connection, WIFI or through the new Wireless Sensor Networks infraestructures that are being in-

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<epcisq:Poll xmlns:epcisq="urn:epcglobal:epcis-query:xsd:1">
  <queryName>SimpleEventQuery</queryName>
  <params>
    <param>
      <name>eventType</name>
      <value>
        <string>ObjectEvent</string>
      </value>
    </param>
    <param>
      <name>MATCH_epc</name>
      <value>
        <string>urn:epc:id:sgtin:0614141.107346.2017</string>
      </value>
    </param>
  </params>
</epcisq:Poll>
```

Figure 5: XML Query for EPCIS Events

stalled in many cities. For a global interconnect, IPv6 addresses are used and IPv4 addresses will be translated into IPv6 if needed to maintain compatibility. The Web Application can also access the same database in order to query, modify or update such data.

2. As non-functional requirements, our application must comply with the following: it seems interactive for the user in order to request information from the server. The interface must be rather simple so that any user can use it. The system should be robust against errors, and report either the time in which they occur or whether not-allowed access to the data is tried out. It must provide security for the data stored in the database, so that they can only be modified by authorized users.

Once these requirements are met, we may enhance them according to the desired approach implemented for our application.

Within this section, it is also necessary to select the platform on which we are going to develop our application. For its development, the following elements have been selected as the proposed architecture is based on Fosstrak EPCIS Repository and we need all the elements are fully compatible with Fosstrak.

Items selected for our development are as follows:

1. J2EE [1] development architecture.
2. Apache Tomcat [8] to be used as application server.
3. MySQL [12] to be used as database administrator.
4. Eclipse [18] and Visual Studio [21] to be used as development environments.

We have selected these elements because they are compatible with Fosstrak and EPCGlobal, they are programmed in Java, they are free, open source and secure. In the case of Visual C#, we use it because SkyeTek development APIs [32] are written in C and Microsoft NET. In this way we connect, configure

and control our hardware SkyeModule M9 [31] and the captured information can be send to the central database for our application.

4.2 Modeling

Once the design is defined and the development platform has been also determined, the next step is to apply to particular cases such as urban transport routing and bus stops location on a map. It also estimates the arrival time of a bus stop.

In our case, our application consists of the design and implementation of an EPCGlobal system based on RFID to store event data, more specifically location events. These data will be available to be accessed through any device and any Web browser.

RFID passive tags are used because they are a good solution for tracking products along the distribution process and they are used in the first layer of our model. The RFID passive tags are cheap, can store up to 128 Kbytes of data and the RFID readers can read several tags simultaneously. Besides, RFID passive tags are used as they do not need maintenance an are more widespread than their active counterparts, so wharehouses, generally, have the requiered handling equipment. However, it is possible to use RFID active tags for specific applications, such as sensors, if required.

After deciding the application, the next step is to create the tables in the MySQL database and the various Java classes needed to work comfortably and easily, with the tables of the database that we create.

Within the database, the first task we carry out will be the filling a table named Read Point. The idea is to place a Read Point at each stop. In this way the name of the stop will be assigned and related to the located data (latitude, longitude and address) to identify it.

The next step will be to create and load the tables in the database that are for working with urban transport lines and the different buses. As our application is rather simple, we only need to define three Java classes to be able to properly write the tables that we have created within the database. The three proper classes are: EPCIS, line and stop. As we can see in Figure 8, the properties of the line and stop classes are specific for the database modeling.

Another important issue to consider when developing the data model, it is to analyze the relationships between the different tables and see how they relate to each other, to achieve a correct and consistent with our application data model. In Figure 9, we can see how they are related and how nodes are connected to.

4.3 Integration

One of the main objectives of regular client-server architecture is the separation of the business logics from the design logics. In other words, the goal is to separate the presentation data layer from the user layer. Currently, when designing an application for big computer systems, it is often programmed via layers or multilevel architectures.

The main advantage of this model of development is that the application is splitted up into several levels or layers. Thus, each layer will play a simple role within the application, so we make a minor

modification on the application, where it is not properly working.

To develop our application, a three-layer model is used (see Figure 6). It will consist of the following layers:

1. Presentation layer, for the user interface of the application. It introduces the system to the user, communicates information to him and returns the answer to the user applying a previous filter in order to check that there is no error. The interface consist of JSP components, because dynamic content is created in an HTML document. the file is completed with JavaScript functions to communicate with Google Maps.

2. Business logic layer, for the business processes of the application. This layer is where the programs are running. It communicates with the presentation layer, to receive requests and present results, and with the data layer, to asking the system manager database for data storage or retrieval.

3. Data layer corresponds to the definition, storage and access to the application data. In an EPC-Global system, it translates into the definition of the tables necessary to store data of events, vocabularies, bus, bus lines, etc. This layer consists of one or more databases management and it receives store request and retrieval information from the Business logic layer.

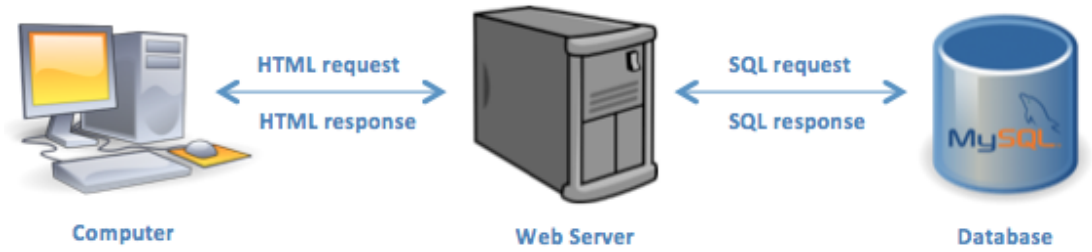

Figure 6: 3-layer model

Once the layer model to be used is decided, the next section will describe the process that we follow for the application development.

Another important task in our development is to connect, to configure and subsequent control of RFID readers with Visual C#. This is a crucial task, because the location information captured is stored in a MySQL database, that is controlled by the developer, and these elements will be located on the map.

Finally, to conclude this section, the main libraries that we have used for deploying this application are:

- SkyeTek .NET API. It is a simple object-oriented interface that allows to use .NET framework in a computer. This interface allows to create commands for SkyeTek Protocol v3 (STPv3) [30] for SkyeModule readers. The requests and responses are written following the STPv3 standard.

- JDBC (Java Database Connectivity) API. This API is necessary to work with MySQL DB. It allows to run different MySQL commands using Java in any DB.

- Fosstrak API. It is neccesary to define an EPCIS Repository to develop our application. For this, the Fosstrak API default libraries, the EPCIS Clients and the TDT Engine are used.

EPCIS Capture Client and EPCIS Query Client. These clients allow capture and query for all the EPC information.

EPCIS Repository. It is a file that is stored in the Web Server. Whenever that the system needs to capture or show information through EPCIS, the system queries to this context. EPCIS Repository has several Java components that are necessary to develop our application.

TDT Engine. It traslates automatically the EPC information[3] that it receives from the Skye-Module M9 into Pure Identity URIs.

- Google Maps JavaScript API v3. With this API, Google Maps can be added in a Web page. Also, we can find several utilities to work and services to add information to Google Maps. Google Maps API v3 is specially designed to work with mobile devices and traditional desktop applications.

5 Location transport and arrival time

Once the theoretical model of the system has been presented, in this section we describe the development of a particular WS for the location of urban transportation. In addition, the Web Application shows more information about bus lines, bus routes, prices, etc.

The WS indicates the arrival time of the bus at a particular stop and it locates all the bus lines on the route map, representing the specific position and displaying the time left for a bus to arrive at a particular stop. Besides, the application can calculate routes between two bus stops with the possibility of one transfer between two different busses.

As indicated in the previous sections, we develop a Web Application based on regular client-server architecture. The client requests through a Web browser the WS to the Web server. In our own local network, there is a DB server that provides access, security and storage of all the data in the system. One of our development goal is that our application can be viewed from any compatible device, using any web browser, and which can be accessed from anywhere in the world.

On the other hand, we will have nodes related to the RFID reader that connects by USB to a micro-processor, so the Web server updates the system when an event occurs. This will happen when an RFID tag goes nearby the antenna of a reader.

Subsequently, the location of the various elements that will form our system through Google Maps will be displayed. As discussed in previous sections, there will be a simulation of the system, due to the unavailability of all the resources involved to achieve the actual implementation.

To begin our development, we create a Web Application information retrieval. To access procedures to the database, it is using the default ones in Fosstrak EPCIS Repository. We will also use all default methods leading to capture EPCIS events, thus following the standard marked by EPCglobal.

In the modeling process of our system, we are designing the necessary MySQL tables (Figure 7) for proper operation. Are also being designed Java classes (Figure 8) which form our model, taking special care in relationships and how to communicate between nodes our architecture consists of (Figure 9).

[3]EPCIS standard works in Pure Identity URI format. The EPC information is a hexadecimal code.

Figure 10 represents a simple Use Case Diagram with the different user options available in the Web Application. In addition, a sequence diagram of a request for a transfer between two different busses is shown in Figure 11.

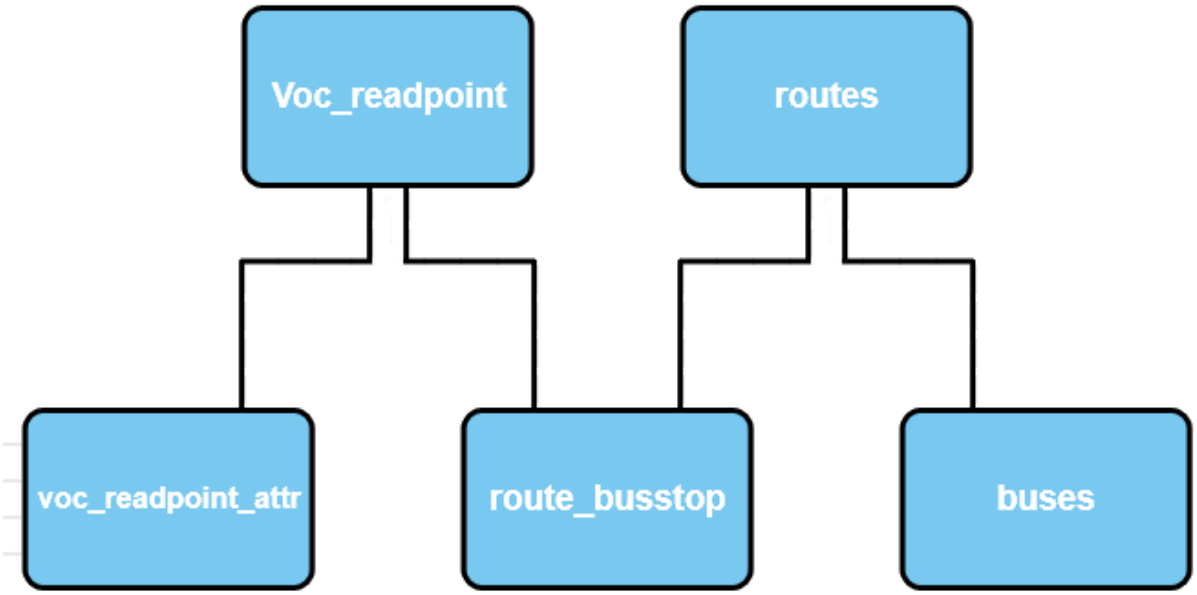

Figure 7: MySQL Tables Diagram

In order to achieve all the information needed to locate all the elements (buses, bus stops, etc.) and calculate the remaining time for a bus to arrive at a specific stop, we are using the SkyeTek libraries .Net API needed to work with the module SkyeTek M9.

Also, to run operations in the database, the JDBC API is being used. Fosstrak EPCIS Repository, as we have already indicated above, being used for capture and information operations.

Finally, the Google Maps JavaScript API is being used in order to use Google Maps and be able to locate buses with great accuracy within these maps. The RFID tag information, that is captured through the antennas, is stored in a database that will be accessed by the application of Google Maps and thus be able to show all that information in the Web Application.

It has also made the connection, configuration and subsequent control of RFID readers through the Visual Studio environment. As mentioned above, this task was crucial because the location information captured is stored in a MySQL database. From this information and through the Google Maps API, we can locate all the buses in our Web Application.

A screenshot of the Web Application is displayed in Figure 12, for showing the route of line C bus and their bus stops on the map, and the time left for reaching the rearest bus stop. As we can see, the Web Application is divided into six different sections with the following options for our application: Index, Schedule and Prices, Routes, Bus Stops, Route Calculation and Waiting Time.

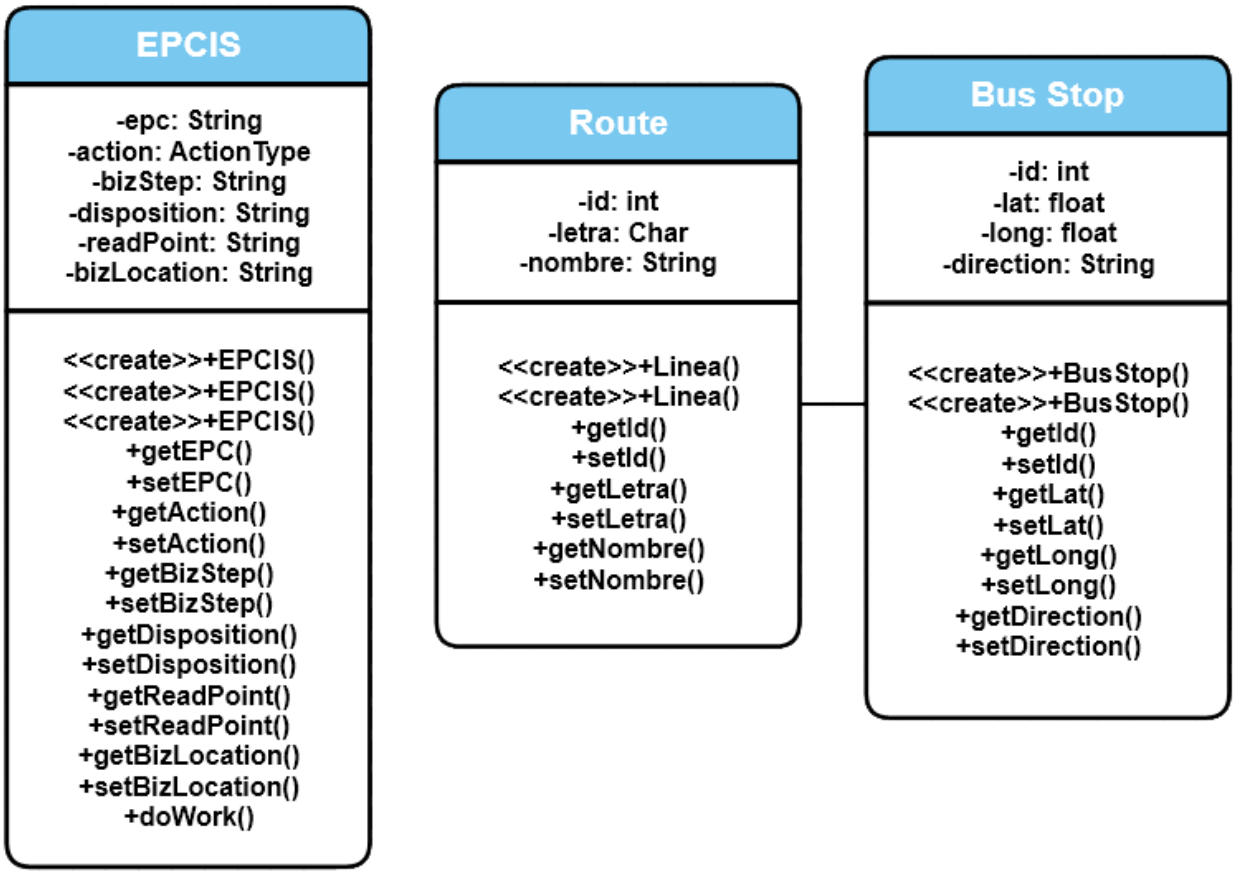

Figure 8: UML Class Diagram

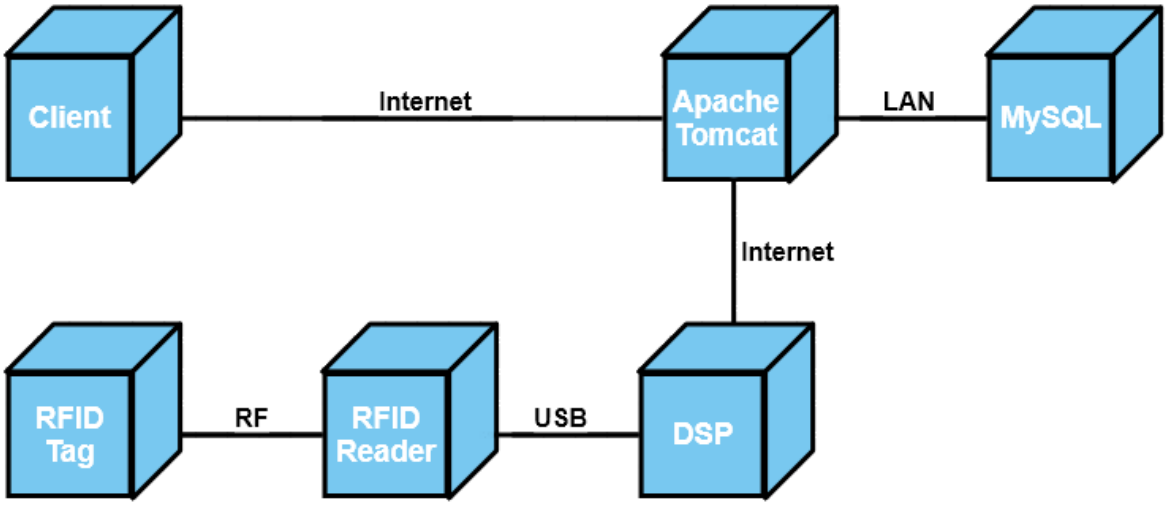

Figure 9: Communication between nodes: UML Deployment Diagram

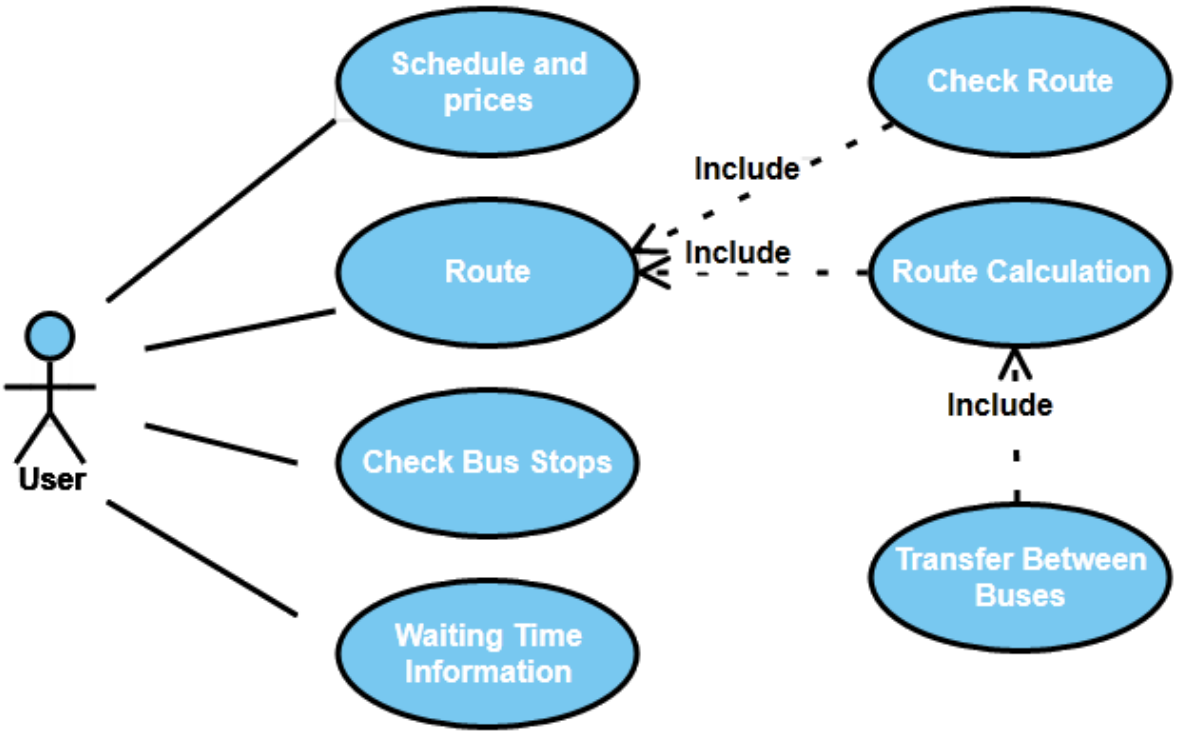

Figure 10: UML Use Case Diagram

- Index. In this section, we briefly explain the Web Application.
- Schedule and Prices. This section includes several information about bus lines, bus stops, phone numbers, points of interest and special bus lines.
- Routes. It show us the different routes of each bus lines.
- Bus Stops. The Web Application show us each bus stop on Google Maps. For that and following the three-layer model, when we select a bus stop, the application sends a query to the DB server and receives the information that it shows us on Google Maps. The Web Application allows to the user filter by bus line to get only the bus stops of one single bus line.
- Route Calculation. This section allows to the user to select an origin bus stop and a destination bus stop, and it calculates the most efficient route with the possibility of one transfer between two different buses. Once the Web Application receives the information, it shows this information (origin bus stop, destination bus stop, transfer between busses) on a Google Maps screen.
- Waiting Time. The last section show us the time left to arrive of a bus of a particular bus line to a certain bus stop. Also, we can get the location of this bus on Google Maps and the route to arrive to the actual bus stop.

6 Conclusions

The creation of a project based on a J2EE Web Application is proposed in this paper. The procedures needed for accessing the databases using the Fosstrak EPCIS Repository to capture events according to

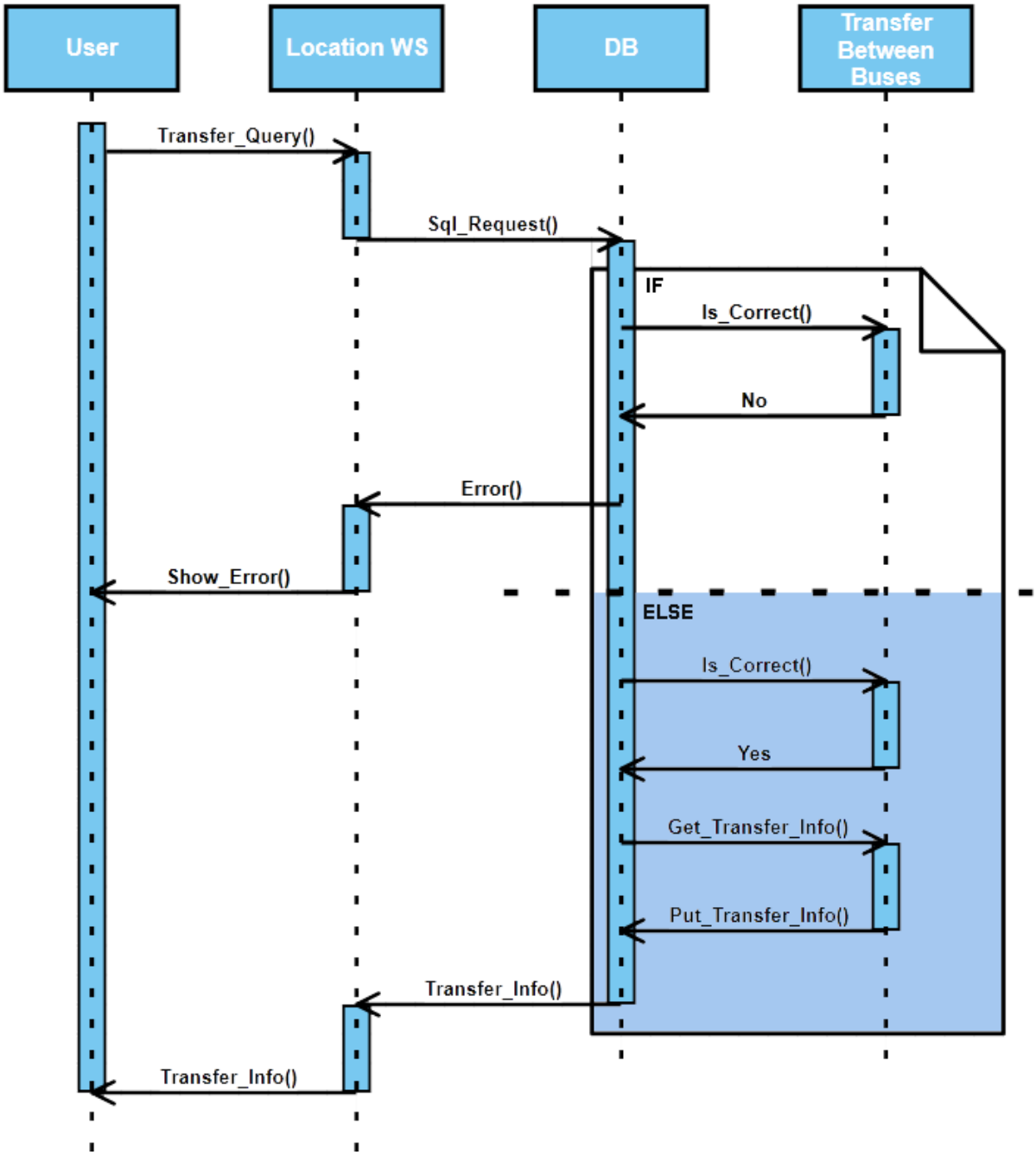

Figure 11: UML Sequence Diagram

the EPCglobal standard have been defined. Although, we have performed small scale tests and we are in negotiations with local authorities for actual testing on a real urban environment.

Combining two technologies, RFID and GIS, the tracking system referred in previous works can be improved and more features can be added, with the design of Web Services. Besides, we have been integrated the tracking system in a Web environment as two Web Services, the monitoring of buses and real-time information of the bus stops.

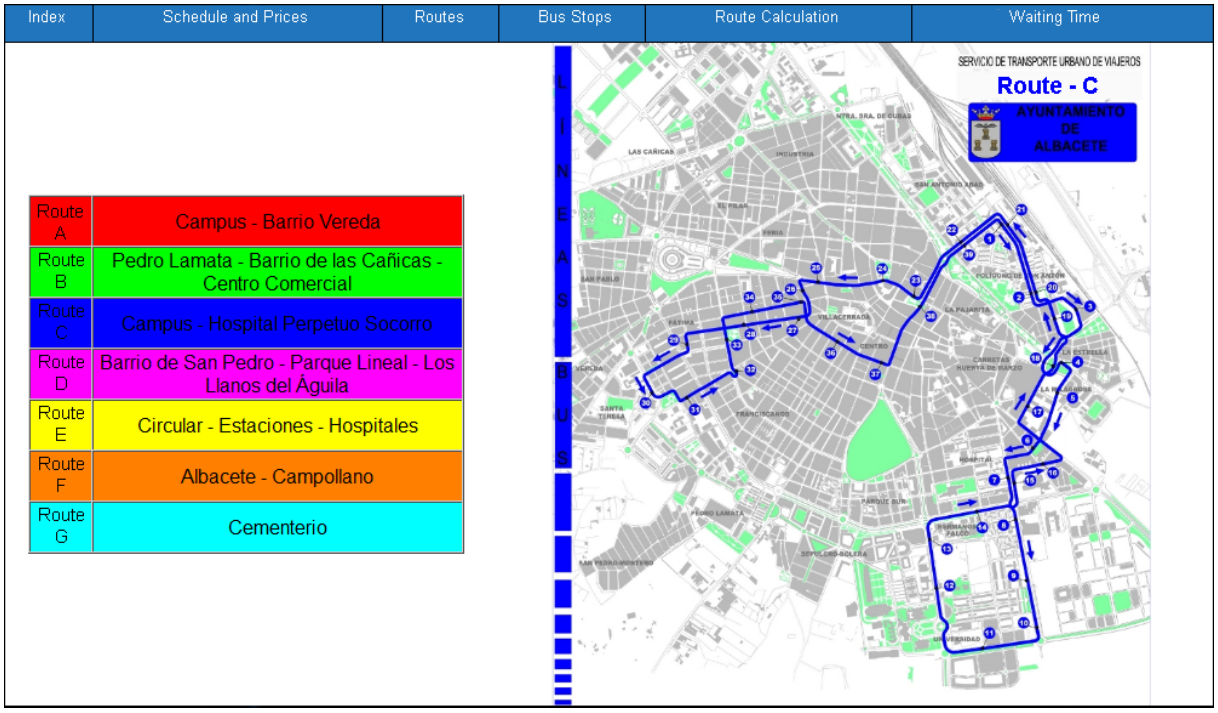

Figure 12: Web Application

In addition, a three-layer model is proposed for the development of our application. Additional similar applications can be developed by using the proposed model. Also we have used JavaScript functions, which may be reusable in future works.

Our original motivation was to complete the development initially achieved by our research group in the lower layers of the EPCIS standard, in order to advance through to the upper layers. Our previous research was focused on developing hardware and networking middleware for RFID systems.

6.1 Discussion

In this section, we will perform a comparison among all the systems mentioned in this paper and ours, indicating the advantages and disadvantages of the RUTMS system. Table 3 shows this comparison in a more graphical way, including the applications we consider important in evaluating RUTMS.

The advantage of our system RUTMS consists on the use of the communications infraestructure of the city itself. RUTMS can transmit data via LAN, WIFI or WSN. In this way, we have got no additional expenses, as in cases where the position must be transmitted from the vehicle, usually via GSM, GPRS, 3G, 4G, etc.

In RUTMS, the installation of RFID readers in each bus stop can initially be more expensive than installing GPS receivers and a mobile device to send the data in each bus. But in our case, we would have an initial investment cost, but we would not have to pay for the communication lines of each bus every month during the usual operation of the system.

	Passenger information display	**APP for mobile devices**	**Route calculation**	**Transfer between buses**	**Number of buses will arrive to bus stop**
London (iBus)	✓	✗	✗	✗	1
Sidney (PTIPS)	✓	✗	✗	✗	1
Auckland (RTPIS)	✓	✗	✗	✗	1
Glasgow (BiAS)	✓	✗	✗	✗	2
BMS using RFID and WSN	✓	✗	✗	✗	1
RUTMS	✓	✓	✓	✓	2

Table 3: Systems comparative

In all systems, passengers are able to query bus information on screens installed at bus stops. Only in BiAS and RUTMS systems, it is possible to check information about the next two buses at a particular bus stop.

Besides the web environment in which we can calculate routes and a transfer between two buses, we are also developing a new app compatible to all the operating systems for mobile devices (Android, iOS y Windows Phone). This application will include the same options implemented for the Web environment we have described previously. These two new contributions we have introduced in the proposed system are not available in any of the related works mentioned in this paper.

One of the disadvantages of the RFID technology compared to GPS: the real-time position of each bus is not available. But, as we have an adequate estimation of the time between stops, the problem is reduced. Ibus, PTIPS, RTPIS and BiAS allow real-time location. While BMS using RFID and WSN and RUTMS only allow estimated location of the buses when not in a bus stop.

Another disadvantage compared to the other systems is that the priority of the bus can not be changed to be higher than the other vehicles. Though only, iBus, PTPS, RTPIS and BiAS have implemented this application. In our system, this feature can be implemented by introducing a new device specifically for this purpose, but this solution will make the proposed system more expensive.

6.2 Future Work

Our future work will consist on developing and integrating several sensors which can send data to a central server so the application data can be stored in the tracking system. Besides, a filtering and collection interface can be used in order to filter and collect raw tag reads, over different time intervals delimited by several events defined by the EPCIS Capturing Application. Besides, as we are considering only one transfer among buses, several transfers can be developed in future implementations of the tracking system.

On one hand, we have considered only one transfer among buses, so several transfers can be developed in future implementations of the tracking system.

On the other hand, a WSN can be added to our system to know the number of passengers on the bus in real time. With this information and the hours of passenger transport peak, each route can be reinforced in their peak times of usage.

Additionally, the date provided by the sensors in the buses to get traffic information in real time can be sent this to other buses or vehicles through WSN.

We are developing several sensors which can send data to a central server so the application information can be stored. Besides, a filtering and collection interface can be implemented in order to filter and collect raw tag reads, over different time intervals delimited by several events defined by the EPCIS Capturing Application.

Our final intention is to develop an architecture and software tools in order to implement tracking and tracing applications in different fields: logistics, people tracking and Wireless Sensor Networks.

Acknowledgments

This work was partially supported by Ministerio de Ciencia e Innovación (project BIA2009-10225) and by Viceconsejería de Ciencia y Tecnología - Junta de Comunidades de Castilla-La Mancha (Project PII1I09-0134-5683).

References

[1] O. Aloquili, A. Elbanna, and A. Al-Azizi. J2ee development frameworks. *Computer*, 38(1):107–110, January 2005.

[2] O. Aloquili, A. Elbanna, and A. Al-Azizi. Automatic vehicle location tracking system based on gis environment. *IET Software*, 3(4):255–263, December 2008.

[3] D. Bajaj and N. Gupta. Gps based automatic vehicle tracking using rfid. *International Journal of Engineering and Innovative Technology*, 1(1):31–35, January 2012.

[4] P. Baronti, P. Pillaia, V. W. Chook, S. Chessa, , A. Gotta, and Y. F. Hua. Wireless sensor networks: a survey on the state of the art. *Computer Communications*, 30(7):1655–1695, May 2007.

[5] A. Burlarley-Hyland, B. Toothill, and A. Das. Use of automatic vehicle location systems, geographic information systems, and global positioning systems to increase efficiencies and decrease cost in maintenance management. In *Proc. of the Maintenance Management Conference (MMC'12), Seattle, USA*, pages 46–54. Transportation Research Board of the National Academies, May 2012.

[6] M. Castro, A. J. Jara, and A. Skarmeta. Architecture for improving terrestrial logistics based on the web of things. *Sensors*, 12(5):6538–6575, May 2012.

[7] K.-T. Chang. *Introduction to Geographic Information Systems*. McGraw-Hill Education, 2010.

[8] V. Chopra, S. Li, and J. Genender. *La Biblia de MySQL*. Wrox, 2008.

[9] R. Clarke, T. Bowen, and J. Head. Mass deployment of bus priority using real-time passenger information systems in london. In *Proc. of the 35th European Transport Conference 2007 (ETC'07), Noordwijkerhout, Netherlands*, pages 1–9. Association for ETC 2007, October 2007.

[10] G. C. Council. The Bus Information and Signalling System - BIAS. `http://www.glasgow.gov.uk/index.aspx?articleid=5203`, 2004.

[11] G. Developers. Google Maps JavaScript v3 API. https://developers.google.com/maps/documentation/javascript/?hl=en, 2013.

[12] P. Dubois. *La Biblia de MySQL*. Ed. Anaya Multimedia, 2007.

[13] J. EberspÄcher, H.-J. Vögel, and C. Bettstetter. *GSM Switching, Services and Protocols, Second Edition*. John Wiley & Sons, 2001.

[14] EPCGlobal. EPC Information Services (EPCIS) version 1.0.1 specification. http://www.gs1.org/gsmp/kc/epcglobal/epcis/, 2007.

[15] EPCGlobal. EPCGlobal Frequently Asked Questions. http://www.gs1.org/docs/epcglobal/Frequently_Asked_Questions.pdf, 2007.

[16] EPCGlobal. The EPCglobal Architecture Framework. http://www.gs1.org/gsmp/kc/epcglobal/architecture/, 2010.

[17] EPCGlobal. Gs1 epc tag data standard 1.7. Technical Report 1.7, GS1 EPCglobal, 2013.

[18] T. E. Fundation. Announcements, Community News, etc. http://www.eclipse.org, 2013.

[19] A. García, Y. S. Chang, and R. Valverde. Impact of new identification and tracking technologies on a distribution center. *Computers and Industrial Engineering*, 51(3):542–552, November 2006.

[20] R. Glidden, C. Bockorick, S. Cooper, C. Diorio, D. Dressler, vadim Gutnik, C. Hagen, D. Hara, T. Hass, T. Humes, J. Hyde, R. Oliver, O. Onen, A. Pesavento, K. Sundstrom, and M. Thomas. Design of ultra low cost uhf rfid tags for supply chain applications. *IEEE Communications Magazine*, 42(8):140–151, August 2004.

[21] M. Gousset, B. Keller, A. Krishnamoorthy, and M. Woodward. *Professional Application Lifecycle Management with Visual Studio 2010*. Wrox, 2010.

[22] M. S. Grewal, L. R. Weill, and A. P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley-Interscience, 2007.

[23] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio. Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. *IEEE Transactions on Services Computing*, 3(3):223–235, July-September 2010.

[24] D. Guinard, V. Trifa, T. Pham, and O. Liechti. Towards physical mashups in the web of things. In *Proc. of the 6th International Conference on Networked Sensing Systems (INSS'09), Pittsburgh, USA*, pages 1–4. IEEE, June 2009.

[25] B. A. Hatem and H. Habib. Bus management system using rfid in wsn. In *Proc. of the 7th European and Mediterranean Conference on Information Systems (EMCIS'10), Abu Dhabi, UAE*, pages 1–8. EMCIS, April 2010.

[26] N. Hounsell, B. Shrestha, and A. Wong. Data management and applications in a world-leading bus fleet. *Transportation Research Part C: Emerging Technologies*, 22:76–87, December 2012.

[27] J. I. S. Jose, J. M. Pastor, R. Zangroniz, and J. J. de Dios. Rfid tracking for urban transportation using epcglobal-based webservices. In *Proc. of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA'13), Barcelona, Spain*, pages 1295–1300. IEEE, March 2013.

[28] A. Mehaffey and G. Jarjees. Preliminary trial results of the public transport information and priority system. In *Proc. of the 8th World Congress on Intelligent Transport Systems (ITS'01), Sydney, Australia*, pages 1–8. ITS America, September-October 2001.

[29] F. E. Project. EPCIS Architecture Guide. https://code.google.com/p/fosstrak/wiki/EpcisArchitectureGuide, February 2012.

[30] SkyeTek. Skyetek c and .net api reference guide. Technical Report 080409, SkyeTek, Inc., 2008.

[31] SkyeTek. SkyeModule M9. http://www.skyetek.com/ProductsServices/EmbeddedRFIDReaders/SkyeModuleM9/tabid/208/Default.aspx, 2013.

[32] SkyeTek. SkyeTek Development APIs. http://www.skyetek.com/ProductsServices/EmbeddedRFIDReaders/Firmware/SkyeAPI/tabid/419/Default.aspx, 2013.

[33] G. Svennerberg. *Beginning Google Maps API 3*. Apress, 2010.

[34] V. Vencatachellum. Auckland's Signal Pre-emption and Real-time Passenger Information System. http://www.ipenz.org.nz/ipenztg/papers/2003/11_Vadi_Vencatachellum.pdf, 2012.

[35] B. Violino. What is rfid? *RFID Journal*, pages 1–2, January 2005.

Author Biography

Jose Ivan San Jose received his Master degree in Advanced Computer Technologies from the University of Castilla-la Mancha in 2011. Currently, he continues his studies as a PhD student and participates in several research projects, and he is a researcher of Institute of Audiovisual Technologies in Cuenca (Spain) since 2010. His research interests are RFID, Web Services, Internet of Things, Smart Objects and Wireless Sensor Networks.

Juan Jose de Dios received the Ingeniero de Telecomunicación degree in 1991 and the Doctor Ingeniero de Telecomunicación degree (PhD in Communications) summa cum laude in 2004 from the E.T.S. Ingenieros de Telecomunicación of the Universidad Politécnica de Madrid (Spain). Since 1999 he has been with the E. U. Politécnica de Cuenca of the Universidad de Castilla–La Mancha (Spain). He is an associate professor of television systems in the Department of Signal Theory and Communications. Previously, he was working for Lucent Technologies (formerly AT&T) at the Transmission Engineering Lab in Madrid (Spain) from 1991 to 1999. His professional interests include RFID technology, image and video processing.

Roberto Zangroniz received the Science (Physics) degree in 1999, and the Electronics Engineering degree in 2001, from the University of Valladolid, Spain. Roberto has ample experience in the design and development of microcontroller driven devices. His research interests include RF data communications applied to RFID, Wireless Sensor Networks and, Real-Time Location Systems through the design of electronics systems. In 2001, he joined the University of Castilla–La Mancha, where he is a Lecturer of the Polytechnic school at Cuenca (Spain).

Jose Manuel Pastor was born in Leon, Spain. He received the Electronics and Automatic Control Engineering degree from the Polytechnic University of Madrid (UPM) in 1991. In 1992 he begins his doctoral research in Robotics in Construction Industry. In 1996 Dr. Pastor moves to the Engineering Department of the Carlos III University of Madrid. He received his Ph.D. in Robotics and Artificial Intelligence in 1997 from the UPM. His research interests include RFID, factory automation, intelligent manufacturing and logistics systems. He is currently an Associate Professor at University of Castilla-La Mancha, and director of Research Institute of Audiovisual Technologies.