**Open Access**

*Open Journal of Web Technologies (OJWT)*
*Volume 5, Issue 1, 2018*

www.ronpub.com/ojwt
**ISSN 2199-188X**

# Consuming Web Data in a Guiding App for Public Bus Users

Miguel Ángel Garrido Blázquez, Paloma Cáceres, Belén Vela,
Carlos E. Cuesta, José María Cavero Barca, Almudena Sierra-Alonso

School of Computer Engineering, Rey Juan Carlos University C/ Tulipán s/n, 28933 Móstoles (Madrid), Spain,
{miguel.garrido, paloma.caceres, belen.vela, carlos.cuesta, josemaria.cavero, almudena.sierra}@urjc.es

## ABSTRACT

*The complexity of urban public bus networks in big cities makes their use very difficult. This paper presents Notify.me, a set of pervasive services for mobility that employs open data from the public bus network in Madrid. Our solution provides both a guiding service to assist users travelling by bus and a notifying service (visual, acoustical and sensorial) that informs them when a relevant point on their route has been reached (transfer or destination). Notify.me needs a starting point, which can be the user's current location, a destination and the preferences regarding the best route for the user. Notify.me requests a route from the Madrid public bus company via SOAP Web services. The back-end responds with the calculated route, the user's route, which includes the bus lines, the transfers and the pedestrian routes needed to reach the destination. Finally, an empirical evaluation of the experiences of users who employed Notify.me is presented.*

## TYPE OF PAPER AND KEYWORDS

Application paper: *Web data processing, Web data consuming, public transport, mobile application, guiding system.*

## 1 INTRODUCTION

The use of public transport may have several drawbacks for users in big cities. In fact, there are several reasons why using the bus can be quite awkward when the user is unfamiliar with the network or has no reference point as regards his/her location. First, the network has a complex structure which has grown with the city, and it uses the city map as a basis – in some respects differently from railways or the underground network, which have often been specifically designed. There are not, therefore, any simple maps to follow: bus routes are described as coloured lines, tangled on a map, on which even bus stops are difficult to identify. Indeed, one of the most useful features, bus transfer, is very difficult to implement when the user does not know whether or not two lines intersect. Moreover, even when a good map is available, travellers often do not know the place to which they are going. Unlike train stations, which are clearly labelled, bus stops can be anywhere: users do not necessarily know whether their destination has been reached.

In summary, it frequently occurs that non-regular users do not know how to use the bus network. They do not know where to go, how to get there, which lines to use, how to combine them, or where to transfer to a different line. It is not just a matter of knowing the right

This paper is accepted at the *International Workshop on Web Data Processing & Reasoning* (WDPAR 2018) in conjunction with the 41st German Conference on Artificial Intelligence (KI) in Berlin, Germany. The proceedings of WDPAR@KI 2018 are published in the Open Journal of Web Technologies (OJWT) as special issue.

direction: indeed, it is more complex than a simple navigator.

This is an area in which ubiquitous and pervasive computing can obviously be of assistance [10], [12]. Bus travellers can clearly benefit from this by downloading maps, or even using the Global Positioning System (GPS) [13] navigator to guess where to go at a particular moment. What is more, mobile ubiquitous computing could be supported with the existing architectural infrastructures by means of Service-Oriented Architectures [15]. SOA, along with the notion of service orientation itself (service-oriented computing, SOC), is currently ubiquitous in everyday software development practice. In fact, this technological infrastructure is available from the Madrid urban bus public company [6]. This company provides any requesting consumer (any consumer that requires it) with an open data platform of its urban bus public network (supported by a SOA) and unlike Google Maps [9], it makes possible to use their bus network data in an unlimited way.

We have therefore, designed a solution denominated as Notify.me in order to assist users in their urban bus movements. It works as a personal assistant: the user asks for a route from a starting point to a destination and the system responds with the calculated route. The starting point can also be the user's current location, provided by the GPS. This calculated route, called the user's route, includes hybrid bus & pedestrian routes, the choice of bus lines, all the necessary transfers, indications of the bus stops traversed, etc. However, Notify.me is not simply a map, as it is able to guide users throughout their journeys. This is done by working with the urban bus network services, by obtaining the current position using the GPS information, and by notifying the user (both acoustically and with visual messages) every time a relevant stop has been reached.

Other route planning applications, such as Google Maps [9] or Citymapper [4], do not offer this kind of notification. Moreover, a multilingual user interface is also provided. Bear in mind that this is different from the classic "navigator" approach, as the user is driven by the bus which follows its standard route, and notifications are, therefore, meant to keep the user on track. As EMT is the owner of bus network data, any event concerning the traffic, bus failures, route changes, etc., will be incorporated to compute every user bus route. To the best of our knowledge, other pedestrian navigational applications do not use such updated information. Notify.me also takes into account an obvious application in the touristic domain: rather than indicating just anywhere, users simply state a place that they wish to go to and the system locates the best route to reach it.

A first prototype of the Notify.me was presented in [3]. This prototype worked offline (that is, unable to be aware of the events happening in the bus network), and used just a subset of the existing bus lines. In any case, Maps and pedestrian routes have been obtained from providers of geographical data, such as Google Maps [9]. The current version works online (including updated information about events) and uses the whole bus network.

In this paper, we present the Notify.me final product, which provides a set of pervasive services for mobility that simplify the user's movements around the urban bus public network in Madrid. The information of the Madrid urban bus public network is requested from the EMT Madrid's Web data platform [7], which returns these data encapsulated as SOAP messages. Notify.me services are currently available only for Android mobile devices. This work has been validated by carrying out an empirical evaluation of Notify.me. As a result of this work, we have transferred the Notify.me technology to EMT, the public bus company in Madrid. Moreover, we have also registered a copyright file in the intellectual property office.

The paper is organized as follows: Section 2 shows the context of the Notify.me solution, while Section 3 shows the architectural infrastructure used to support our solution. In Section 4, we show how Notify.me was developed, and this development is validated in Section 5 by carrying out an empirical evaluation. Some of the lessons learned from this are also shown. Finally, in Section 6 we present our conclusions and show some open issues.

## 2 THE CONTEXT

Notify.me is a pervasive solution that facilitates users' mobility around the urban bus public network by providing guidance and notification services with a multilingual user interface. The Notify.me user requests a user's route from the system by indicating a starting point and a destination, and Notify.me then starts working. On the one hand, it guides public transport users around the urban bus network of a city by following users' routes, by identifying their movements and positions from GPS information and by showing this information on a map in real time; and on the other, it sends a notification to users when they have to change bus or when they arrive at their destination. Users can also select their preferred user interface language.

One of the most significant terms in the context of Notify.me is the user route. In Figure 1, we attempt to explain what a user route is by means of an example. A user route is composed of a starting point (represented with the solid black circle) and a destination

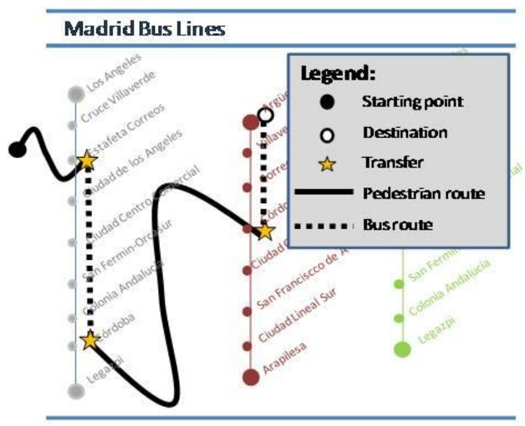


**Figure 1: Example of a user's route**

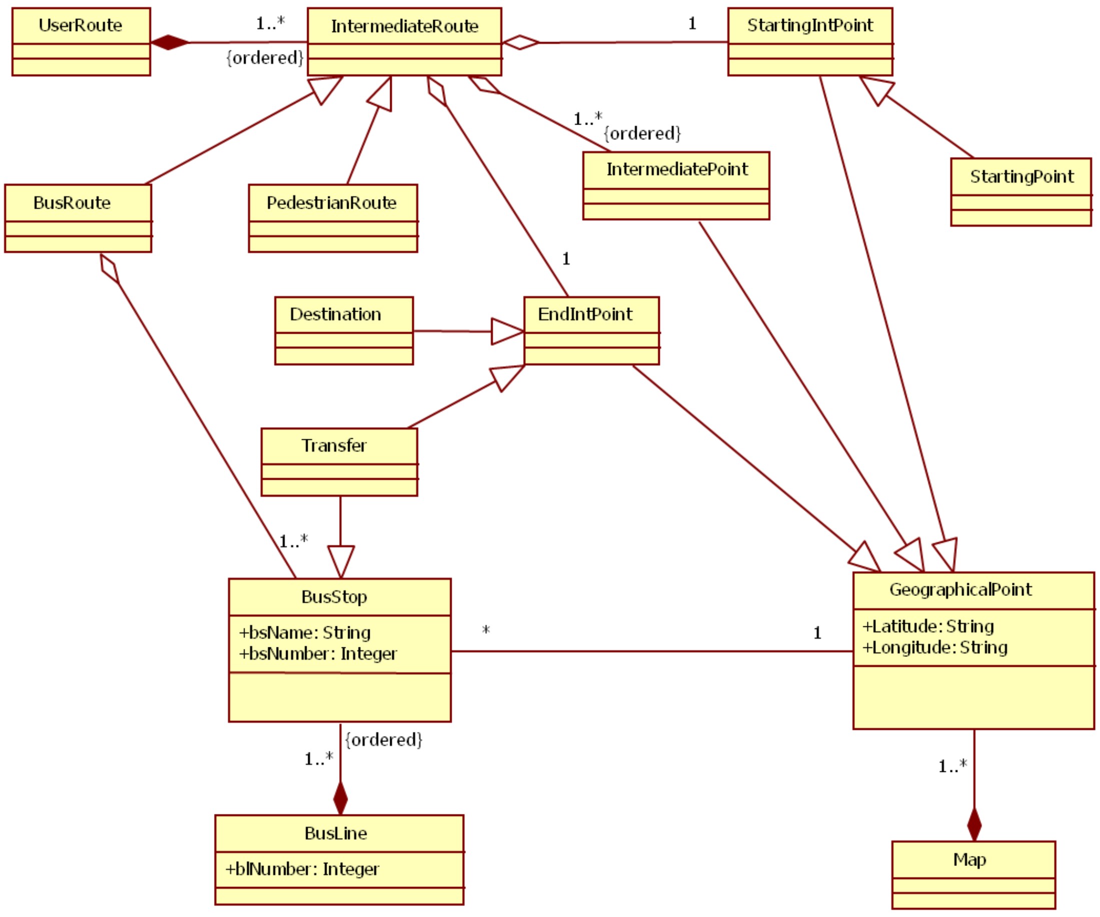


**Figure 2: Notify.me domain model**

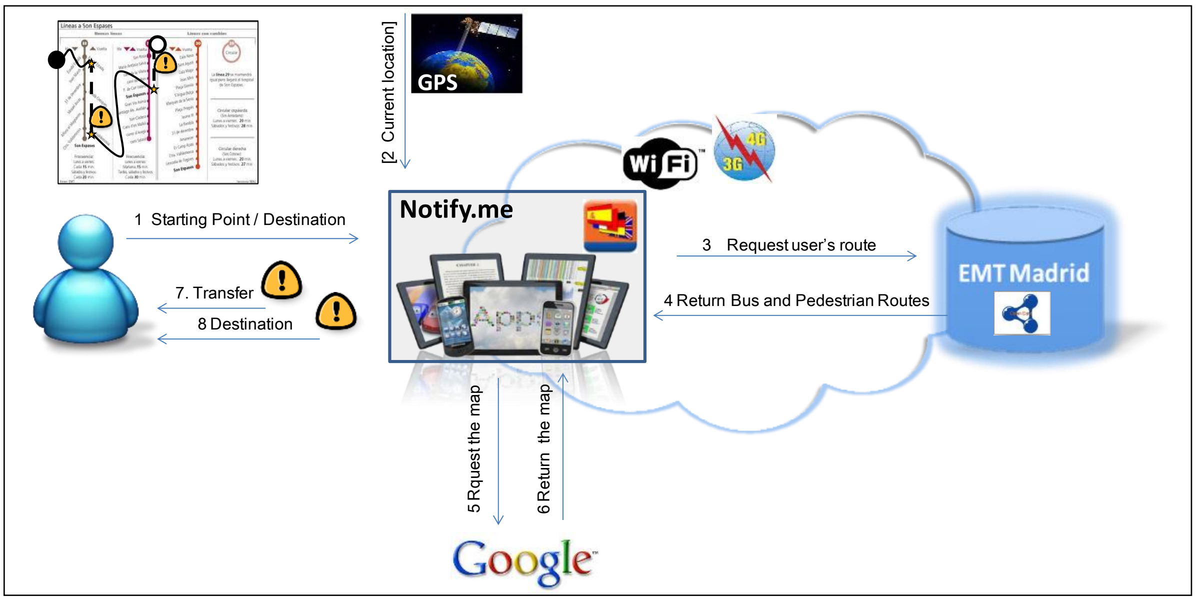


**Figure 3: Notify.me behaviour**

(represented with a white circle). The user's route is a set of intermediate routes, represented with the black lines (both dotted and solid); an intermediate route can be a pedestrian route, represented by means of the solid lines, or a bus route, represented with the dotted lines; a transfer, represented by means of a star, is a point at which an intermediate route ends and connects with another intermediate route. In order to identify the relationship between a user's route, the pedestrian and bus routes, and the geographical coordinates, we have defined the domain model represented by means of a UML class diagram (see Figure 2).

As will be observed, a UserRoute is composed of different IntermediateRoutes. An IntermediateRoute can be a PedestrianRoute or a BusRoute. An IntermediateRoute always has three or more ordered elements which are a unique StartingIntPoint, one or several IntermediatePoints and a unique EndIntPoint. Each one is a GeographicalPoint with its latitude and longitude attributes. The StartingPoint (that is, the user's route Starting Point) is a specific StartingIntPoint. The StartingPoint is a place, which can be an address or a POI, but it is also possible to select the current position as the starting point of the route which will, in this case, be located by the GPS. An EndIntPoint can be the Destination (the user's route destination) or a Transfer (a place where the user could change from one bus line to another). In Figure 3, the destination can be identified as the white circle and the transfer as a star. The Destination is also a place, which can be an address or a POI. A Transfer is a point located at a bus stop. A BusLine is composed of different BusStops and each BusStop is a GeographicalPoint.

Notify.me must be in accordance with this model if the user is to be provided with the guidance and notification services in an adequate manner. Figure 3 shows the behaviour of Notify.me. As will be observed, the user indicates the starting point and the destination (see message 1). It is also possible to select the current position as the starting point of the route (see message 2). In this case, Notify.me then requests the user's route from EMT Madrid (see message 3), which then returns the user's route according to the starting point and the destination (see message 4); Notify.me subsequently requests a map, which Google Maps returns (see message 5 and 6). Notify.me integrates the user's route, including the bus stops on the bus routes and the pedestrian routes, with the map obtained from the provider of the geographical data. From then on, Notify.me guides the user while travelling along the user's route: it shows the map and the user's route on it and it represents the moving user's current location, provided by the GPS, redrawing it when necessary. Finally, Notify.me sends a notification to the user when s/he has to change bus to complete his/her route (see the notification triangle in message 6) or when the user is approaching his/her destination (see the notification triangle in message 7).

We have included Notify.me‘s whole behaviour (the language selection and the guiding and notifying services) by identifying two specific use cases, as will be seen in the use case diagram in Figure 4. The first use case, called Select your preferences, permits the language in the user interface to be selected by the user actor. The second, called Guide me, is the main use case, which provides a specific user's route with which to

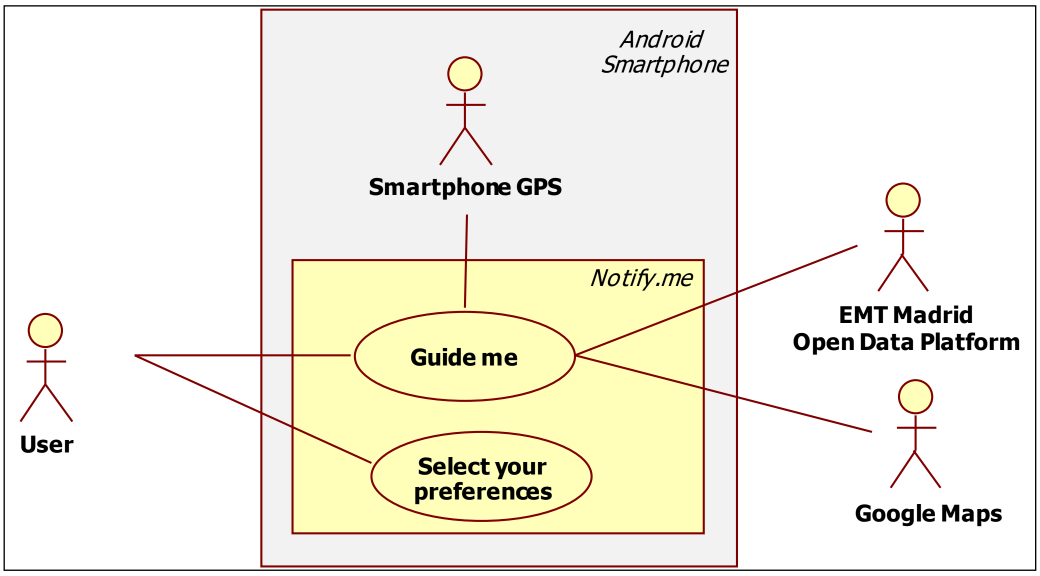


**Figure 4: Use Cases Diagram of Notify.me**

travel through the urban bus network. This use case has four actors: the User, the Smartphone GPS, Google Maps as the Geo Data provider and EMT Madrid. The initiator actor is the User, who will indicate both the starting point and the destination of his/her route. The starting point may sometimes be the user's current position, which will be provided by the GPS. The GPS of the smartphone is the actor, which also provides the geographical coordinates along the user's route. Google Maps is the actor, which provides the layout map in order to show the streets of Madrid (and to depict the user's route in the user interface of the Notify.me application). EMT Madrid provides the data concerning the user's route in Madrid.

It is relevant to emphasize how Notify.me works with the EMT Madrid service interface to obtain data related to bus routes in Madrid. The architectural infrastructure that makes it possible to access open data via Web services and SOAP messages is shown in the following section.

## 3 Architectural Infrastructure

As mentioned previously, the concept of Service-Oriented Architectures (SOA) is currently ubiquitous in the present reality of software, and it is not therefore necessary to insist on its relevance. The concept was conceived only a decade ago, and during this period it has extended to cover the widest possible range. There is seldom any aspect of software engineering (from development environments to distributed systems, and its current role in database technology should also be emphasized) that has not been affected or influenced - even decisively - by its presence.

SOA has been discussed at great length during this last decade, and it is currently possible to find at least three different definitions of it:

1. Any software architecture that considers the notion of Web Services (WS) as central, and which is typically based upon this technology (i.e. "a SOA"),
2. The specific architectural style that follows several of the relevant standards, particularly the SOAP protocol and WSDL interfaces – and, therefore, a strict subset of the former group (i.e. "the SOA"),
3. Any software architecture to which the use of the generalized notion of service is central, does not necessarily use the WS technology, and can be on a larger scale – this includes both the first group and also SaaS-based architectures.

The Notify.me architecture can be simply defined as a SOA of the first group; i.e. it considers that the notion of service id central, independently of the technological details and the different variants of the concept of WS. In particular, this service-oriented architecture uses the original definition of service technologies (i.e. WS standards, using the SOAP protocol). The Notify.me architecture contains WS-* services when using the EMT Madrid service interface.

Independently from the underlying technology, there are several features inherent to the definition of SOA and, therefore, derived from its core nature. These features imply that any SOA is, a priori, more dynamic and flexible than most other architectures particularly in the case of "traditional" component-based architectures.

Thomas Erl [8] defined this set of features as a set of eight service-orientation principles, namely: (1) the standardization of service contracts, (2) loose coupling, (3) service abstraction, (4) service reusability, (5) service autonomy, (6) service statelessness – with some nuances, (7) service discoverability and (8) service composability. Several authors have proposed variants and extensions of this list, but it can serve as a first conceptualization. There are at least two features (also implicit in these principles) that we would like to emphasize, since they have a large number of consequences. These are the presence of external composition mechanisms and their intrinsically open architecture.

With regard to the former, we must take into account that services are always part of a modular system – they are not conceived to be used independently, but as part of a larger system. But, unlike previous approaches, which tend to work at compile time, services are designed to be composed at runtime. This is not a trivial difference: we are not able to assume that a service knows the rest of the system, nor that each dependency has been resolved a priori – i.e. that each component has a well-defined place.

It is, therefore, necessary to separate the interface of services (i.e. their API) from the core of the service itself, which is consequently isolated from the rest of the system. Services are not conceived as a part of a composite; instead, once they are deployed and operational, they are included in a compound system, defined a posteriori, which implies a bottom-up approach. The two existing service composition models (orchestrations and choreographies) are, therefore, essentially external composition models. This means that the essence of any service architecture is within the architecture itself (either centralised in the service orchestrator or distributed in a decentralised choreography). Service applications, and even simpler service mashups, are external composition models.

Another consequence of this composition model is that it becomes crosscutting. Service encapsulation, unlike traditional models, does not forbid the same service from belonging to several architectures simultaneously – every composition is orthogonal.

With regard to the second feature, i.e. openness, there are of course many other open systems, which are not necessarily service-oriented. In practice, an open system is any system that uses a standard interface and, therefore, admits the composition of any external client, sometimes even blurring its specific features. However, services are defined from the start in a flexible manner, offering a specific feature (function) and guaranteeing a certain level of quality (QoS). The open and dynamic nature of service compositions, which is derived from these two features, makes it possible to present them as the "next step" in the evolution of software aggregates. During the last decade, Software Engineering has dedicated a great amount of effort to software architectures, as the main expression of existing software compounds, and with an emphasis on component-based approaches. But current service models describe completely dynamic architectures in which new unforeseen elements can join the system, and the original elements can even leave it, in which the structure can easily change or be combined with another in an orthogonal manner; i.e. in which any element (service) can join the structure on a temporary basis, and can leave it at any moment, or remain within it indefinitely, according to its own necessities and requirements.

In summary, service-oriented architectures currently define the most flexible kind of computing system that can be built; and this, when superimposed on many other useful features, such as pervasiveness or the potential for mobility, is the reason why our platform is built and conceived on top of this underlying infrastructure. Notify.me has been designed to work within a SOA infrastructure that includes orchestrated services, i.e. external composition mechanisms.

## 4 OUR SOLUTION

As mentioned above, the main contributions of Notify.me are a guiding and a notifying service with a multilingual user interface. The main features are enumerated below. For the guidance service, Notify.me has to:

- Operate with bus lines and pedestrian routes from the open data platform of the urban bus public network provided by EMT Madrid.
- Include POIs or an address as a starting point or destination. The starting point can also be the user's current position
- Work with the API provided by Google Maps.
- Use the GPS to locate the user's current position.
- Generate SOAP messages to request the user's route and interpret those messages upon their return.
- Show a map and the user's current position in real time on that map to guide the users.
- Provide textual and audio directions to enable users to get to a place.

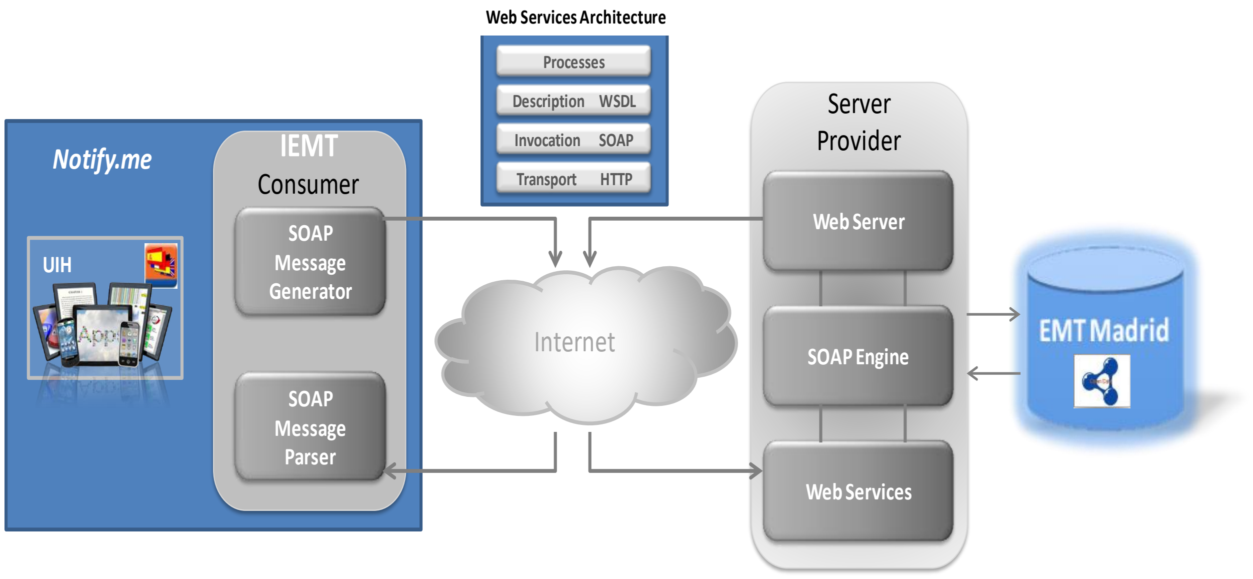


**Figure 5: Solution as regards accessing open data via Web services**

For the notification service, Notify.me has to:

- Advise both when the destination has been reached and when the user has to change bus line.
- Show a textual message, beep an acoustical signal and ensure that the mobile device vibrates.

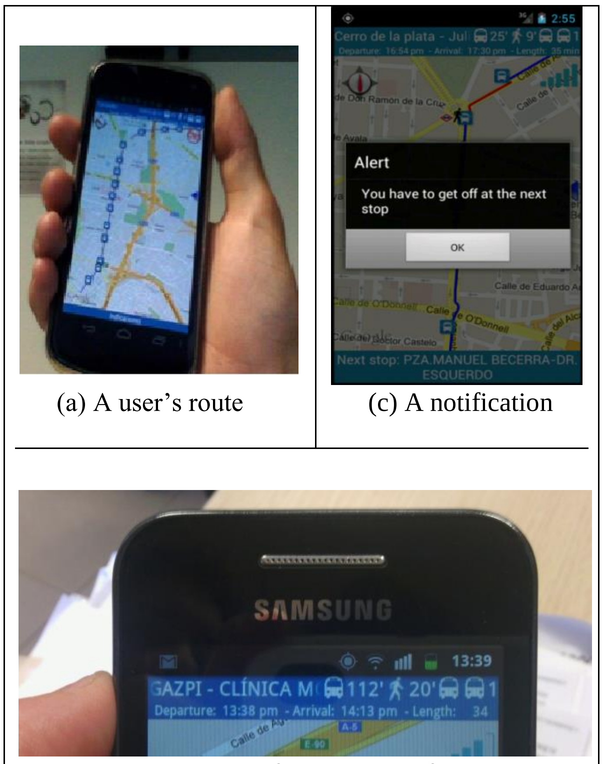
Notify.me must provide several interfaces with which to support these two services: an Interface to work with Location Services and the GPS of the mobile device, called ILS; a User Interface with the Human actor (UIH) with which to request data; another Interface to interact with Google Maps (IGM); the Interface with which to access open data from the EMT Madrid (IEMT). With regard to ILS, the API for location services in Android (android.location) is used to work with the user's current location and the GPS of the mobile device. This API contains classes and interfaces that define Android location-based and related services. We principally work with android.location.LocationManager.

With regard to UIH, we have designed some simple screens on which users can select the starting point and destination of their routes. Users can additionally see the path they have already followed on their route while moving by bus or on foot, and they can also select their preferences (i.e. the language).

With regard to IGM, Notify.me works with the Google APIs add-on, which includes a Maps external library, com.google.android.maps. The key class in this library is MapView, a subclass of ViewGroup in the Android standard library. MapView displays a map with data obtained from the Google Maps service. When the MapView has a focus, it can capture keypresses and touch gestures to pan and zoom the map automatically. It also provides all of the user interface elements necessary for users to control the map. Notify.me, therefore, uses the class methods to draw Overlay types on top of the map. In general, the MapView class provides a wrapper around the Google Maps API that allows our application to manipulate data from Google Maps by means of class methods, and makes it possible to work with Maps data as one would with other types of Views. The Maps external library is not part of the standard Android library and may not, therefore, be present in some compliant Android-powered devices. Similarly, the Maps external library is not included in the standard Android library provided in the SDK. The Google APIs add-on provides the Maps library, thus allowing users to develop, build and run map-based applications in the Android SDK, with full access to Google Maps data.

With regard to IEMT, SOAP WS is used to request the most suitable user's route as regards the user's preferences (i.e. the fastest route). It is first necessary for users to state their starting point and destination. These are then translated into the right format in order to subsequently request a user's route from EMT Madrid via SOAP-based WS. EMT Madrid returns an XML file which contains the data needed to obtain the user's destination. It is then necessary to parse the data and interpret them in order to, on the one hand, provide the user with understandable information, and on the other, provide guidance and notification services.

Figure 5 basically shows the main elements needed to communicate with the EMT Madrid service interface. We work in a Web service basis infrastructure. As will



Alert

You have to get off at the next stop

OK

Next stop: PZA.MANUEL BECERRA-DR. ESQUERDO

(a) A user's route

(c) A notification

(b) Top of the user interface

**Figure 6: Notify.me screens**

be seen in the Web Service Architecture stack, the data is requested by generating a SOAP message following the WSDL specification provided by EMT, by means of a Web Service via HTTP. This signifies that Notify.me is the consumer and EMT Madrid is the provider. In order to carry out the complete communication, Notify.me must first generate a SOAP message by invocating one of the provider's Web services. EMT Madrid then serves a data XML file from the open data platform in a SOAP message and Notify.me receives it. Notify.me next parses the message and interprets the information to show it to end users and to integrate it onto the map provided by Google maps.

*Notify.me* app has been implemented by means of Eclipse [4] with the Android Studio and SDK tools [1]. The third is a plug-in for the Eclipse IDE that extends the capabilities of Eclipse, thus providing interesting utilities such as a graphical environment in which to develop the graphical interfaces of apps for Android, tool integration, custom XML editors, etc. We should also stress that we have considered some of the best practices from [14], which have been adapted for this development.

Figure 6(a) shows a photo that represents a bus route on which each bus stop is drawn as a bus icon. Figure 6(b) shows the top of the user interface in which we can see: in the first row, the user's route name composed of the starting point and the destination names; how long it will be necessary to travel by bus and to walk on foot in minutes and, finally, how many changes it will be necessary to make to reach the destination; in the second row, the time of departure, what time it is necessary to leave to get on the bus and the length of the route. Figure 6(c) shows a text alert message regarding getting off the bus.

## 5 Empirical Evaluation of Notify.me

It is well known that software evaluation is necessary if a useful and user friendly application is to be attained. In particular, the evaluation of interactive systems requires focusing on the user interface. There are various kinds of evaluations as regards user participation in the software evaluation. Analytical evaluations do not consider the user's participation, whereas empirical evaluations do. As Notify.me offers mobility services for users, we considered it necessary to take the user's participation into consideration. We have carried out an empirical evaluation using a questionnaire based on that of Brooke [2].

The questionnaire included ten questions, as shown below, and had to be evaluated with a mark between 1 (strongly disagree) and 5 (strongly agree):

Q1. I think that I would like to use Notify.me frequently.

Q2. I found Notify.me unnecessarily complex.

Q3. I thought Notify.me was easy to use.

Q4. I think that I would need the support of a person with technical knowledge to be able to use Notify.me.

Q5. I found that the various functions in Notify.me were well integrated.

Q6. I thought there was too much inconsistency in Notify.me.

Q7. I would imagine that most people would learn to use Notify.me very quickly.

Q8. I found Notify.me very cumbersome to use.

Q9. I felt very confident when using Notify.me.

Q10. I needed to learn a lot of things before I could get going with Notify.me.

In order to support the evaluation, we provide to users a questionnaire, the Notify.me app to be downloaded [42], a help video with a brief explanation (2–3 minutes) of its characteristics and its operation, and a form concerning the gender, the age and the device most frequently used by the user who answered the questionnaire. We carried out the evaluation using the following procedure: first, users downloaded the app and used it while travelling around the Madrid urban bus network;

**Table 1: Results of the users' evaluation**

| Category | Subcategory | Number of users | Percentage |
|---|---|---|---|
| Gender | Male | 50 | 49,02% |
| | Female | 52 | 50, 98% |
| Age | 17-24 | 30 | 29,41% |
| | 25-39 | 28 | 27,45% |
| | 40-55 | 26 | 25,49% |
| | > 55 | 18 | 17,65% |
| Device most frequently used | Smart devices | 87 | 85,29% |
| | Others | 15 | 14,71% |

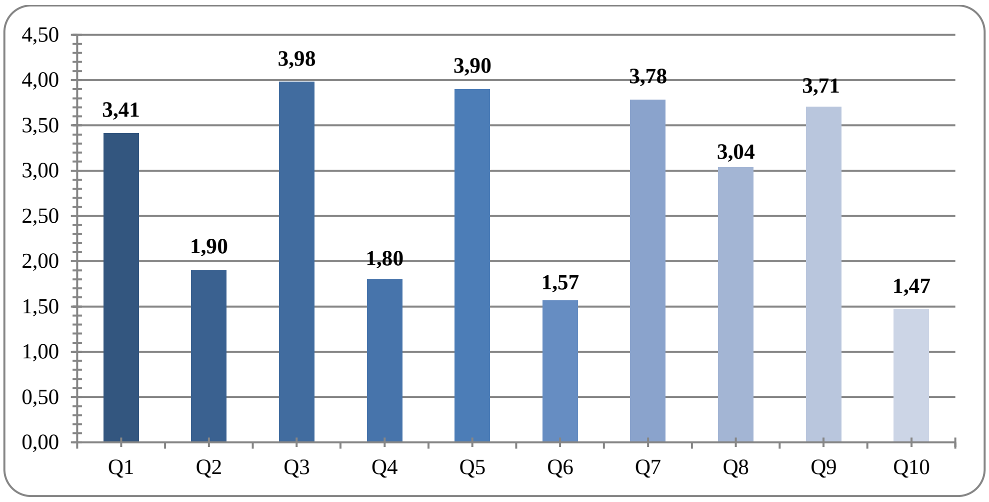


**Figure 7: Average mark per question**

the users then answered the questionnaire and filled in the form; and finally, we analyzed the data collected as regards Notify.me usage.

102 users answered the questionnaire related to the experience. With regard to age, we have established the following intervals: between 17 and 24; between 25 and 39; between 40 and 55; and over 55. We have also considered whether the devices most frequently used by the user are smart devices (smartphones, tablets, etc) or another kind of device (PC, mobile phone, etc.). In order to depict the information compiled, Table 1shows the number of end users and the percentages regarding gender, age and the device most frequently used on a daily basis.

With regard to the average mark per question (see Figure 7), Q8 is the worst evaluated (below the average mark). The reason for this might be that Notify.me has neither online help nor contextual messages. Moreover, it was necessary to write the starting point and the destination street, taking into account uppercase and lowercase letters and a comma before the street number.

Figure 8, shows the average data for gender and age. We have established eight data sets: M1, F1, M2, F2, M3, F3, M4 and F4, in which the gender is M (Male) and F (Female), and the age intervals are 1 (17 to 24), 2 (25 to 39), 3 (40 to 55) and 4 (over 55).

As will be observed in this figure, the worst evaluation is made by older women (F4). There is a tremendous difference between their answer marks and those of the other people surveyed. In fact, they had difficulties in familiarising themselves with the application and we observed that they felt uncomfortable using it during the evaluation test. The greatest difference in their marks with regard to the others concerns question 5 (Q5). This may be because this question takes into account technical characteristics.

In general, young people (M1 and F1) gave better marks than the others in all the questions. This may be because younger people like new technologies in general. They were very optimistic and enthusiastic about using Notify.me. However, M2 gave the worst

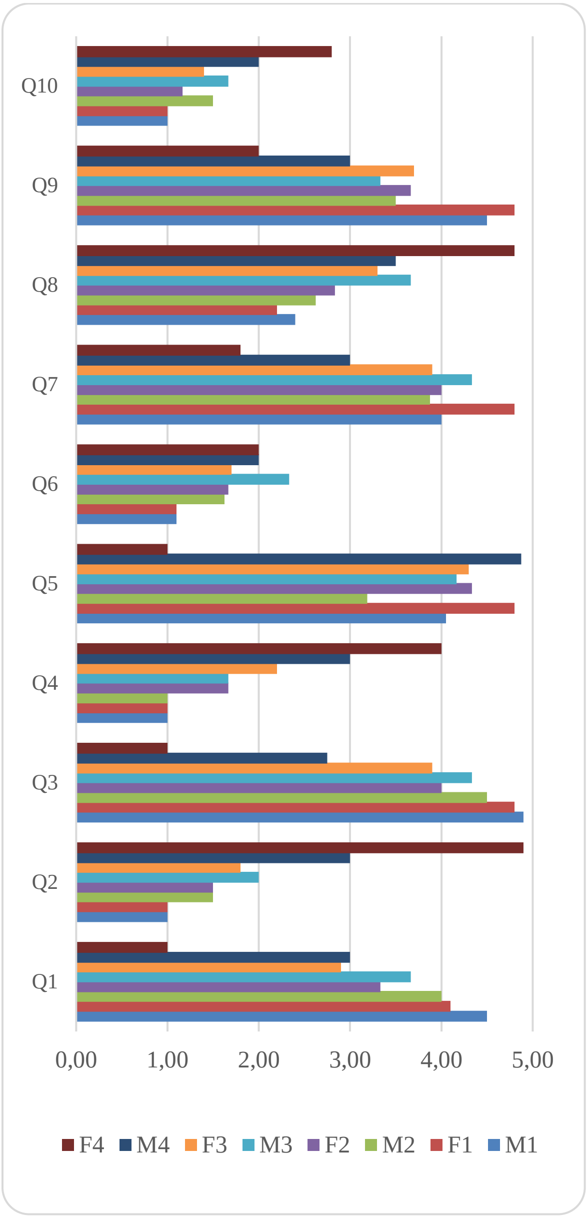


**Figure 8: Average mark by age/gender**

mark for Q5. We observed that they are more critical than the rest with regard to the presentation of the map on the smartphone screen, as there was sometimes a delay before it appeared.

Figure 9 shows the average marks per question and device. In general, people who usually use smart devices are more demanding of Notify.me, and it is for this reason that the best marks correspond with the people who use this kind of devices. The exception is Q8. The reason for this might be the same as that mentioned above, that is, Notifye.me has neither online help nor contextual messages. Moreover, it was necessary to write the starting point and destination street, taking into

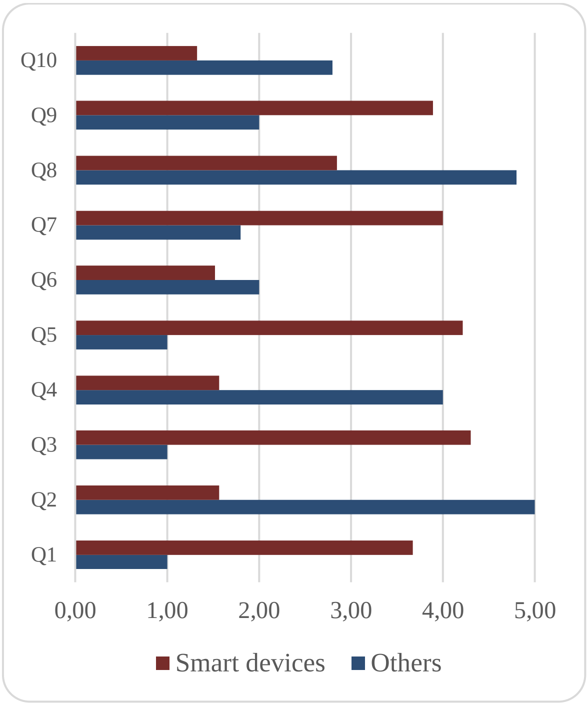


**Figure 9: Average marks for used devices**

account uppercase and lowercase letters and a comma before the street number.

## 6 LESSONS LEARNED

After developing and evaluating Notify.me, we identified some important aspects that we wish to improve in future developments in order to guarantee that Notify.me behaves correctly. As mentioned throughout the paper, Notify.me offers users two specific services. One of these is a guidance service through the public bus network, following the user’s route. This user’s route can be a hybrid of bus & pedestrian routes. With regard to pedestrian routes we should point out that in some cases, the pedestrian route guides the user to the middle of a road or to cross anywhere. Notify.me does not, therefore, guide users in an accessible manner, and we must work to improve this.

With regard to the notification service, we wish to guarantee that users get off the bus at the correct stop. The notification is usually send to the user when the location indicated by the GPS goes past the last bus stop but one. But, what happens if the GPS turns off during this period? For example, what happens when the bus is running through a tunnel? In this situation, if the space in time without a GPS signal is greater than the time taken to reach the destination bus stop, the notification will not be send on time and the user will not get off the bus. Therefore, we must study how to carry out a simulation during the time that the GPS does not work to ensure that Notify.me works correctly. One possible solution might be to use stored pre-calculated routes, bus stops and times.

Notify.me shows the user's route on a map, on which the user's current position is also drawn. As the user (or rather the user's mobile device GPS) is moving along the user's route, the user's current position is moving on the map. This signifies that the map is constantly changing in relation to the user's current position in real time following the previously calculated user's route. One problem with the first approach was that the user cannot move backwards and forwards on the map to see previous or subsequent places, bus stops, etc., and some users have shown interest in this. Therefore, an improvement must be made to permit the user to move backwards and forwards on the map.

## 7 Conclusions and Future Work

In this paper, we have presented a pervasive solution that will facilitate users' mobility within the public bus network in a big city. Our solution offers: a guidance service to assist users to travel around the public bus network in urban areas; a notification service to indicate the next stop at which to get off the bus, both for transferring and for getting to the destination; and a multilingual user interface.

This idea arose because it can be very complex to obtain complete knowledge about a bus network in a big city. Tourists, non-habitual travellers, or residents of the city itself who live in different districts do not know, or only know a small part of, the network and may consequently experience difficulties when travelling by bus because it is not always easy to orient themselves within the network.

We have, therefore, designed Notify.me in order to help these bus network users to travel using this means of transport without needing to know the network: On the one hand, Notify.me offers the possibility of indicating a starting point and a destination to the user, and provides a map with the user's route and the current location via GPS. When the user is moving during the journey, the user's route varies depending on the current location. When the user is approaching a transfer point or his/her destination, Notify.me alerts him/her by sending a textual message and an audible signal, and the mobile device vibrates.

Notify.me has been created to use the open data platform of the urban bus public network of the Madrid public bus organization (EMT Madrid), which provides data accessibility via SOAP WS. We have evaluated Notify.me by means of an evaluation test as regards its perceived ease of use. We chose a sample of 102 users, who travelled around Madrid on the urban bus network using our services.

As future work, we are developing a new version of Notify.me which will provide the user with a more complete application with improved pedestrian routes and POIs, with a user interface in German and Chinese, and with a route planner.

We wish to stress that this work is being carried out at the University of Rey Juan Carlos with collaboration from the Madrid public transport bus company, EMT Madrid, in the framework of the Chair of Ecotransport, Technology and Mobility at the same university.

## Acknowledgements

This work has been developed in the framework of the CoMobilityproject (TIN2012-31104), Multiply@City project (TIN2016-78103-C2-1-R) and the Access@City project (TIN2016-78103-C2-1-R), funded by the Spanish Ministry of Economy and Competitiveness.

## References

[1] Android, “Android studio and SDK tools home page”, https://developer.android.com/studio/, accessed 18th May 2018.

[2] J. Brooke,“SUS-A quick and dirty usability scale”. *Usability Evaluation in Industry*. Vol. 189, no. 194, pp. 4-7, 1996.

[3] C. E. Cuesta, P. Cáceres, B. Vela and J. M. Cavero, “CoMobility: A Mobile Platform for Transport Sharing”. *ERCIM News*, no. 93, 2013.

[4] Citymapper, https://citymapper.com, accessed 17th July 2018.

[5] Eclipse Foundation, “Eclipse home page” http://www.eclipse.org/, accessed 18th May 2018.

[6] Empresa Municipal de Transportes de Madrid, “EMT Madrid home page”, http://www.emtmadrid.es/, accessed 18th May 2018.

[7] Empresa Municipal de Transportes de Madrid, “Open data platform of EMT Madrid”, accessed 18th May 2018.

[8] T. Erl,*SOA: Principles of Service Design*. Prentice-Hall, 2007.

[9] Google, “Google Maps”, https://maps.google.es/, accessed 18th May 2018.

[10] H. Mügge, K-H. Lüke and M.Eisemann, “Potentials and Requirements of Mobile Ubiquitous Computing for Public Transport”. In:

*Proceedings of the UbiLog Workshop at Informatik*, vol. 110, pp. 237-246, 2007.

[11] Notify.me, "Notify.me web site", http://www.vortic3.com/notifyme/, accessed 6th April 2014.

[12] J. Park, H-C. Chao, T. Shon and M. Denko, Guest editorial, "Theme issue on smartphone applications and services for pervasive computing", *Personal and Ubiquitous Computing*, vol. 16, pp. 611-612, 2012.

[13] U. S. Air Force, "GPS specifications", http://www.gps.gov, accessed 18thMay 2018.

[14] W3C, "Mobile Web Application Best Practices", http://www.w3.org/TR/mwabp/, accessed 18th May 2018.

[15] W3C, "Web Services Architecture", https://www.w3.org/TR/ws-arch/, accessed 18th May 2018.

## AUTHOR BIOGRAPHIES

**Miguel Ángel Garrido Blázquez** obtained an MSc in Computer Science from Rey Juan Carlos University (Madrid, Spain) in 2014. He has worked as a computer technician in several entities for 10 years and is currently an assistant professor and a Computer Science PhD student at Rey Juan Carlos University. His research areas are web engineering, model driven development, semantic web, linked open data, transport and accessibility.

**Paloma Cáceres García de Marina** obtained an MSc in Computer Science from the Polytechnic University of Madrid (Spain) in 1993 and a PhD in Computer Science from Rey Juan Carlos University (Madrid, Spain) in 2006. She is an associate professor at Rey Juan Carlos University. Her research interests are software engineering, web engineering, model driven development, data science, open and linked data, semantic Web and transport and accessibility. She is the author and co-author of several national and international papers related to these areas.

**Belén Vela** has an MSc in Computer Science (Carlos III University) and a PhD in Computer Science (Rey Juan Carlos University). She is currently an Associate Professor in the Department of Computing Science, Computer Architecture, Programming Languages and Systems and Statistics and Operative Investigation at Rey Juan Carlos University (Madrid, Spain). She belongs to the Vortic3 Research Group. Her research interests are focused on Software Engineering, Information System Engineering, Data Science, Open Data, Databases (NoSQL), Model Driven Development, Transport, Accessibility, Scientometrics. She has had numerous papers published in prestigious journals and conferences.

**Carlos E. Cuesta** is an Associate Professor of Software Engineering at Rey Juan Carlos University (Madrid, Spain). He has a PhD in Information Technologies (2002) from the University of Valladolid. He belongs to the VorTIC3 Research Group. His main research area is Software Architecture, including such topics as Self-Adaptive Systems and Systems-of-Systems, and relative to such fields as Concurrency Theory, Formal Methods, Agreement Technologies, Intelligent and Distributed Systems or Data Engineering. He is currently the Program Chair of the 12th European Conference on Software Architecture (ECSA 2018). He has published in premier conferences and journals, such as the International Journal of Information Technology and Decision Making, or Future Generation Computer Systems.

**José María Cavero** has an MsC in Computer Science from the Polytechnic University of Madrid and a PhD in Computer Science from Rey Juan Carlos University. He is an Associate Professor at the School of Computer Science (Rey JuanCarlos University). His research interests are focused on Databases, Open Data, Transport, Accessibility and Scientometrics. He has had several papers published in prestigious journals and conferences.

**Almudena Sierra-Alonso** received her Ph.D. in Computer Science from the Universidad Politécnica de Madrid, in 2000. She is an associate professor in the Computer Science and Statistics Department of Rey Juan Carlos University. Her research interests are focused on software engineering, data science, open and linked data, semantic Web and, transport and accessibility. She is the author and co-author of a several papers related to software engineering, semantic web and teaching in Engineering Education (in areas such as operating systems and software engineering).