

Mobile Architecture for Communication and Development of Applications Based on Context

L.M. Soria-Morillo, J.A. Ortega-Ramírez,
L. González-Abril, and J.A. Álvarez-García

Computer Languages and Systems Dept
University of Seville
41012, Seville
`{lsoria,jortega,luisgon,jalvarez}@us.es`

Abstract. The arrival of ubiquitous computing and the increasing use of mobile devices can get a lot of information about the user. This information is used by some applications to adapt its functionality to the user itself and the surrounding environment. In this way the applications need to become more autonomous and less each time user interaction. However, the computational cost, battery consumption and the complex process of obtaining useful information from sensory data means that many applications do not use this information in a massive way. This research proposes a framework and a middleware for the development, implementation and communication of contextual applications. Using this architecture allows communication between applications so that they can share applications without contextual information should both generate. Thanks to definition of a SOA platform, subscription to services provided by other applications is possible through the middleware.

1 Introduction

In the last few years, the Context Based Computing has undergone a major revolution, mainly due to three reasons: the integration of sensors into mobile devices, presence of increased generation of mobile phones in “the daily lives of users” and the massive connectivity between devices and Internet.

On the one hand, technological progress has allowed the integration of a large number of sensors in next generation mobile devices. Years ago, a large infrastructure would be needed for assembly. Examples of these sensors are triaxial accelerometers, chips based on GPS positioning, wireless connectivity features like WiFi and Bluetooth as well as light sensors and more.

Furthermore, the integration of mobile devices in the daily lives of users of a population, allows the development of useful applications for users that can be used anywhere thanks to the portability of devices where run on. We should note that, due to we are working with portable elements, energy savings must be present in any development takes place, as it will be necessary to maximize the usage time of devices. Thus the batteries time is a problem. In addition, it is necessary maximize battery time for the convenience of users of devices, since a high recharge rate can reach displeasing them.

The applications launched by use of context, is a step forward in computing. In such applications, the device itself is which launches applications based on specific conditions of some environmental variables and state of the user who wears it. By obtaining, analyzing and processing data from sensors and minimizing the interaction between the user and the device itself is possible to speed up the use of the device.

2 Objectives

The core motivation of this research is to solve the high degree of dispersion when developers create common implementation of applications based on context and no author to date has defined a framework that is fully adapted to the development of ubiquitous systems standard from a common view for all mobile applications.

The creation of this framework would be a great advantage for applications to exploit the advantages of knowledge of the context surrounding the user to increase the functionality of applications. In addition to increased functionality, most important would be the total adaptation of content presented to the user and the state which is both the user and his/her environment at the time of execution.

In addition to creating a standardized framework, the capabilities of the work environment must support the integration of all context recovery techniques in the most ubiquitous device known: the mobile. This should be done at any time maximizing the number of contexts recognized by the framework, so that the potential of the development of applications is as high as possible.

Under this development framework and application structuring, is going to be developed a communication middleware between applications. Thus, based on our own ontology for contextual information systems, applications can share all the necessary information. This drastically reduces the complexity of building new contextual applications, because information could be provided by another application that already offers this functionality.

3 Scenario

The following is a usage scenario of the development system based on context. We will analyze the interaction of Daniel, a user of next generation mobile device that has acquired an application developed with the exposed framework, and which uses the proposed middleware.

Are 6:30PM and makes a beautiful summer night. The system detects that Daniel is at home, thanks to indoor positioning system and labeling of common sites. During all the night, the mobile phone has been in eco-battery mode, because thanks to the activity recognition module, the system knows mobile phone has been in the table without any use. The system detects that Daniel has left home, thanks to detection of exit to outdoors, and automatically the own system activates the GPS device. The exit detection module is not implemented by own application, but this one a request to the middleware to search if necessary information is available. After consulting the information, the middleware will provide it to the application, in this case if Daniel has left or not the location where he was.

It's Saturday, the day in which Daniel, with his girlfriend Mar often go to the cinema to see films released the previous day. Therefore, after getting into the car, the detection module determines that the frequent destinations most likely fate of Daniel is the home of his girlfriend Mar and activates the navigation system with the specific purpose to meet the traffic to the destination. Once together, Mar and Daniel go to the cinema.

Already in the cinema, Mar meets her sister, but since they do not routinely uses the Bluetooth device, she has disabled it. This makes the subsystem responsible for detecting proximity using Bluetooth cannot find any matches. However, Mar starts talking to her sister about what they will do next weekend. Thanks to the module of detection by voiceprint, Daniel's device recognizes the voice of his sister-in-law Rosa and detects her proximity.

Through a search process on the agenda, the device detects that Daniel has something yet to talk to Rosa, that is, has a to-do with her. Due to the system has detected that Daniel is close to her, shows an alert with this task. Because the film does not start until in half an hour, Daniel and Rosa have decided talk about, so the agenda of Daniel is released from the appointment with Rosa.

4 Related Works

The first researches related to ubiquitous computing emerged in 1991 from the hand of Mark Weiser [2]. This work set a precedent in how to understand computing, and brought the benefits of information systems to the daily lives of users, i.e. what we now understand by ubiquitous computing.

During the next years, the trend of ubiquitous systems went almost exclusively the calculation of the position and its applications. This was aided largely the launch of GPS (Global Position System), which reached full operational capability in 1995, and the creation of portable receivers for this technology.

During the years 2000 and 2001 a series of frameworks for the management of ubiquitous systems were developed [6], [7] and [8]. In particular, the most widely used architecture of the systems of the time was ContextToolkit. This tool allowed integrating several sensors in one system so that the addition of a new sensor could be carried out in a fast and efficient without being necessary to reprogram the entire system. However, the main limitation of the system was that applications could not obtain context data from ContextToolkit, which was faced with the philosophy of applications based on context that years later adopted the scientific community.

However, the real revolution of ubiquitous computing has been in the past three years due to the increase in integrated sensors on mobile devices. Since we mentioned earlier in the Introduction section, massive use of mobile phones has encouraged the development of applications based on context. A clear example of this type of application is the work of Jae Sik Lee et al [10]. It implements a music recommendation system based on the date, location, season, day of week, average temperature, maximum and minimum as well as weather conditions at the site and the present moment. Thus, the system is able to determine the area where the user is and play the music the user usually plays in a context similar to the current.

In the last years has appears a lot of framework mainly related with internet of things and collaborative environments [1]. However mobile phones middlewares are not yet totally defined and there is no works related to this field in the references, at least in an integrated manner.

5 Context Technologies

After entering the work we carry out, this section shall proceed to give the basic ideas on which we based the system to be developed. In this section we will make a clear distinction between the various technologies used to support the attainment of the context in which the user is at any time. These technologies will be determined by the device on which is executed the framework, so that not all devices support all functionality. However, the technologies that will be described throughout this section are included in the next-generation devices that are currently marketed.

Any system that is based on the context to run needs to know a set of information to model the environment around the user. One hand, this model allows taking decisions of execution and on other hand, learning across decisions taken by the user in a given context as a basis for possible automatic decisions. By tradition, as presented above, it is necessary to know who performs the action, where it takes place, when it occurred and what action has been taken. The following technologies enable us to get answers to all these questions. Furthermore, the system will add new information that traditional systems have not taken into account.

5.1 Activity Recognition

The activity carried out by the user also has great value to determine the current execution context. When we speak of activity, we refer to the action that is carried out the user from the analysis of the values obtained from accelerometer sensors.

By learning and classification of a number of parameters obtained from the accelerometer device, the system is able to classify the activity that the user is performing. We do this out using a classification system based on probability tables and decision trees, and a second level, governed by a hidden Markov model (HMM). In this respect we have developed a detection method for classifying activities with a success rate of 95% the activity takes place. In addition, this system is possible to add new patterns of activity to the system without involving reprogramming of the platform.

5.2 Outdoor Location

At present there are very precise location systems outdoors. The most widespread of these is the GPS tracking system. This system requires a direct view of satellites to function properly, so indoor, this system is invalid.

Despite this drawback, global coverage and accuracy make this system the most widely used worldwide. However, the main drawback of this system is the high energy cost GPS receiver. This is especially important in embedded systems in mobile devices. Therefore, at this point, the work will focus on reducing energy consumption of this system in order to extend as far as possible, the battery useful time between charges.

A detection system to external inputs and outputs has been developed. This makes it possible to disconnect the GPS device when the user is indoors and reconnect it when the user goes out. This causes the batteries of mobile devices increasing use time by 230% between charges.

5.3 Indoor Location

The current techniques for indoor positioning are useful in sensorized spaces, since most of them are based on RFID tags [3], bar-codes, QR-codes or proximity sensors to determine the user's position on a roof or the proximity to a particular object. However, this type of positioning techniques is uncomfortable since it requires an infrastructure for its operation. There are other projects that base their position in the triangulation of WiFi networks. The main problem of such solutions is that WiFi access points must be of high quality in order to eliminate fluctuations in signal strength. In addition to this we must bear in mind that in areas where the density of WiFi networks is low, the effectiveness of the system is not assured.

The aim of this work in the field of indoor positioning is getting a new positioning techniques based solely on accelerometer, digital compass and other sensors included in next generation mobile devices. Thus, it is possible to obtain an approximate user's position with a margin of error as small as possible anywhere roofing.

In contrast to RFID-based techniques, the solution to be proposed in this work can be used anywhere without the building is adapted for positioning. This is important when we speak of a ubiquitous system. Using this technique will be possible to positioning the user at any house, office or shopping center where he/she is.

5.4 People Detection

Detection of people near the user opens a range of possibilities for context-based system and therefore the functionality of the framework that we are developing will increase. A detection system makes the application knows at all times who around the user. The system will be even capable of storing information of new people who had not previously interacted with the system's user.

For detection of people close to the user we will use two different but complementary technologies, detection based on sound fingerprints[9] and detection based on Bluetooth signals.

5.5 Semantic Context Induction

Mainly due to poor security of the traditional methods of storing information collected from the user's position, it is necessary to develop a new model. This model is based on the labeling of storage locations instead of latitude and longitude of the location of the user.

The labeling of places introduces much semantic information in the storage system, so that, only the user and the individuals within their environment could understand the information.

6 Architecture Layers

Once studied the various functional modules supported by the system, it is time to analyze the layers that make up the framework itself. Applications developed based on the layers proposals, will be able to perform all necessary tasks identified in the applications based on context. Furthermore, each layer consists of one or more modules, which contain specific functionality for specific software developed needs. Obviously, each module is interconnected with other modules in other layers of the architecture, so it should be abundantly clear the responsibilities of each.

Figure 1 shown de architecture of the framework with its seven layers.

As we shall see when each layer is look in depth, the main advantages of this architecture is on the one hand, the independence of the particular specifications of the sensors used and on the other hand, the ability to share information with other applications and even other information systems present on remote machines or devices via data exchange modules.

On the other hand, we must pay attention to the architecture proposed in this chapter provides the ability to launch applications based on the certain characteristics of the environment, denoted as context-shot applications.

As stated above, we then describe in more detail each of the modules in the architecture for the development of applications based on given context.

6.1 Sensor Adaptation Module (SAM)

This part of the structure contains all the logic for adjusting the data from the sensors of the device or other applications through the middleware. Generally, the data must be processed prior to the application can proceed with the analysis, classification and use of such data.

6.2 Information Classification Module (ICM)

The information from sensors or other system applications is not usually enough to provide functionality to the application to develop, but these data must be processed to generate new information. It is precisely this layer of the architecture which provides such access to developers.

ICM layer is one of the main modules of the architecture; it is where information is processed by algorithms and processing methods. This layer will have support of SAM layer and storage ISM layer through which the application can obtain and store persistently all the information generated.

6.3 Information Storage Module (ISM)

The purpose of this layer is to provide a method of persistent or temporary storage of all information generated by lowers layers of the application. This layer should provide methods to access, modify and delete information stored. On the other hand, we should avoid as much as possible to other layers of the architecture to access the ISM, except for the ICM and SSL, because usually the information stored in context has a high degree of privacy. It is therefore essential to prevent other applications or systems uncontrolled access to such information.

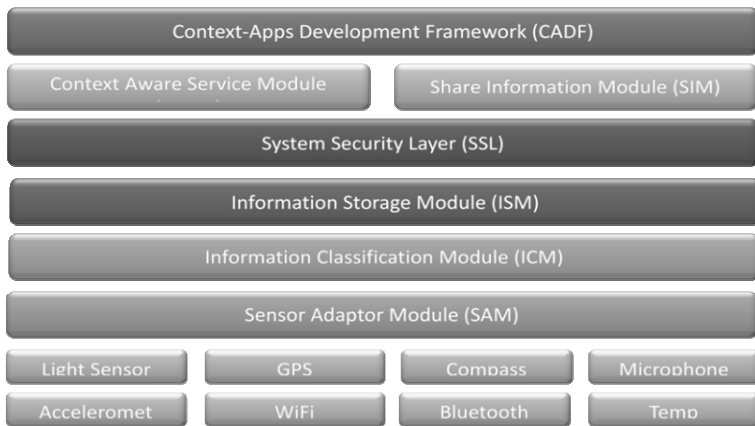


Fig. 1. Framework architecture diagram

6.4 System Security Layer (SSL)

The security layer provides an effective system to prevent access to information by applications without permission to. In this section does not specify specific security methods to prevent access, although it is recommended that access to the main database of the application is done safely. In addition, whenever possible, they have to delegate the storage of information to the middleware, which has a specific module for this purpose. Thus it is possible to isolate and wean the application of data stored. In this way, the information can be accessed or stored using the functions that the middleware offers.

6.5 Share Information Module (SIM)

As mentioned above, access to persistent or temporary information stored in the database should be limited to applications with sufficient privileges to do so. In addition, the information stored must be marked as available for a specific application profiles to provide greater security to the system.

Furthermore, external applications residing in other systems may access information through web services, so as to facilitate the exchange of information with other systems. This task will be done by SIM module, which will enable the application designed to send and receive information from other systems via Internet.

6.6 Context Aware Service Module (CASM)

This module allows storing information about the launch of applications based on specific context. In particular, the information stored will be the application to execute, the execution parameters and context that must be met for execution.

Thus, the CASM module communicates directly with the middleware module sending said registration information of an automatic release. In addition, this module

provides a point of information exchange in case the current application should be executed because another application has registered the process of automatic execution.

7 Middleware

In this research a middleware following SOA architecture is going to be generated, which allow applications to provide services to other applications. In addition, a certain application can subscribe to the services offered by another easily.

The proposed middleware consists of five layers. In Figure 2 is shown a diagram explaining the different modules of the middleware architecture.

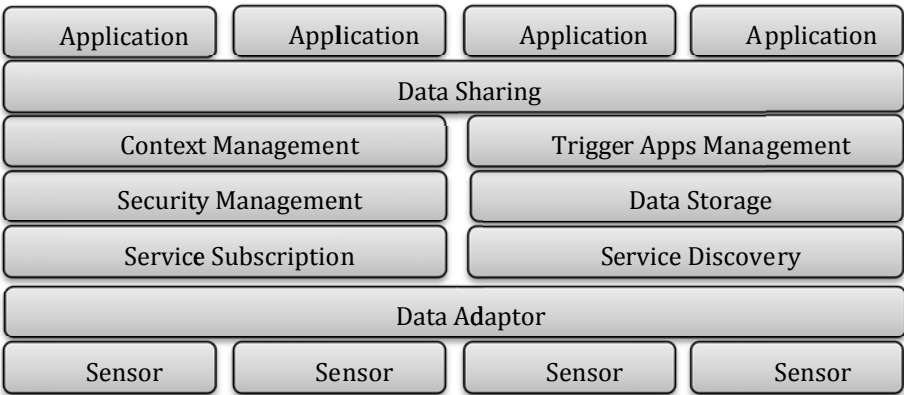


Fig. 2. Middleware architecture preview

The first of these is the adaptation layer of sensory data. It will package the information received from sensors and will be introduced to the middleware as a service, so that both the information generated by other applications such as collection from the sensors, will be accessible from any application that requires it. Secondly we have the data storage and security system. The first of them is responsible for storing all the necessary information in a persistently, so that information is available at any time for that application that need it. As has been seen, this module should implement a policy of deleting data, to prevent malfunction due to excessive memory. Security manager will be responsible for determining whether an application can do or not to use a particular service.

The third layer is subscription services module and service discovery module. Although both are described later in detail, we must take in mind that these modules are responsible for managing all events generated by the middleware for sending and receiving context information between applications. The service discovery module is designed to provide applications a way through which indicates whether a particular service is or is not available in the system. Instead, the subscription to the service

module is used by applications to register the notification system in a particular service that has already been checked that is available.

At the next level of middleware context manager appears. This module is the intermediary between applications and the middleware itself, providing an interface to access the system. In addition, process all requests coming from the applications themselves and delegate the service module that is appropriate. On the other hand, the Context Management unifies all communication logic modules, standardizing the ontology intended to be communicated. In the same level as this module is the launching applications module, which is responsible for receiving requests and performs automatic launching of applications.

Finally appears the information exchange module, which aims to facilitate communication between applications that are not hosted on the device itself, using a channel of communication and external dissemination middleware. This module is an expansion point in the architecture, because it can extend the operation of the system to a collaborative environment in which applications are not always implemented on the device itself, but in a distributed environment.

8 Future Works

Although middleware and architecture development stage is still early, some objectives can be defined to ensure proper adaptation of architecture to the realm of contextual and ubiquitous applications. In this way should be complete the identification of sources of information compatible with the architecture so that it succeeds in obtaining an ontology compatible with existing information sources and provide a general enough to support news data that have not been taken into account.

On the other hand it should provide a set of optimal solutions for each of the requirements in the layers of architecture. This will get applications that use this architecture using compatible technology, which will facilitate the reuse of modules among several applications at the time of development.

9 Conclusions

The result of this work will aim to solve the traditional problems of systems based on context and provide a common development environment for most devices. This ensures the independence of individual sensors and provides a development interface for applications.

The framework will make the development of applications based on context is conducted in a much more efficient than they have done so far. This will ensure an increase in this type of application, with the comfort this means for the user.

Finally, we must comment that the structure of the framework will make allow the developers to extend the functionality of them. On this way, the developer could adapt the framework to their own needs, thereby increasing the performance, quality and minimizing the time cost of application development based on context.

Acknowledgements

This research is partially supported by the projects of the Ministry of Science and Innovation INCARE (TSI2006-13390-C02-02) and ARTEMISA (TIN2009-14378-C02-01).

References

- [1] Song, Z., Cárdenas, A.A., Masuoka, R.: Semantic middleware for the Internet of Things. In: Proceeding of Internet of Things, IOT (2010)
- [2] Weiser, M.: The computer for the 21st century. *Scientific American* 265(3), 94–104 (1991)
- [3] Hameed, B., Khan, I., Dürr, F., Rothermel, K.: An RFID based consistency management framework for production monitoring in a smart real-time factory. In: Proceeding of Internet of Things, IOT (2010)
- [4] Schmidt, A., Beigl, M., Gellersen, H.-W.: There is more to Context than Location. *Computers & Graphics Journal* 23(6) (1999)
- [5] Schmidt, A., Forbess, J.: What GPS doesn't tell you: determining one's context with low-level sensors. In: Proceedings of the ICECS 1999, 6th IEEE International Conference on Electronics, Circuits and Systems, ICECS 1999 (1999)
- [6] Dey, A.K., Abowd, G.D.: The Context Toolkit: Aiding the Development of Context-Aware Applications. In: Workshop on Software Engineering for Wearable and Pervasive Computing, Limerick, Ireland, June 6 (2000)
- [7] Dey, A.K., Abowd, G.D.: CybreMinder: A Context-Aware System for Supporting Reminders. In: Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing (HUC2K), Bristol, UK, September 25-27, pp. 172–186 (2000)
- [8] Dey, A.K.: Understanding and Using Context. In: *Personal and Ubiquitous Computing*, vol. 5(1), pp. 4–7. Springer, Heidelberg (2001), <http://www.personal-ubicomp.com/> ISSN: 0949-2054
- [9] Lu, L., Zhang, H.-J. (eds.): Unsupervised speaker segmentation and tracking in real-time audio content analysis, vol. 14. Association for Computing Machinery, Inc, New York (2005)
- [10] Lee, J.S., Lee, J.C.: Context Awareness by Case-Based Reasoning in a Music Recommendation System. In: *Ubiquitous Computing System*, pp. 45–58 (2008)
- [11] Nord, J., Synnes, K., Parnes, P.: An Architecture for Location Aware Applications. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002), vol. 9 (2002)