Journal of Multimedia Information System: ISSN 2383-7632(Online)
VOL. 1, NO. 1, September 2014, pp. 11-22

UbiqBIOPARC: A Wireless and Sensor Based Context-Aware System for an Enhanced Guide Experience

Jose-Vicente Sorribes[1], Juan-Carlos Cano[*, 1], Carlos T. Calafate[1], Pietro Manzoni[1]

Abstract

This work discusses and evaluates the use of wireless and multi-sensor based technologies to develop UbiqBIOPARC, a new generation zoological park that has been created based on the zoo-immersion concept. It offers appropriate contextual information to zoo visitors, depending on their preferences and the environment in which they are positioned. It combines the flexibility of the iPhone SDK, the connectivity provided by 3G technologies, the location capabilities of GPS, and the orientation offered by a digital compass integrated in the device. In this document the overall architecture and the implementation steps followed to create this context-aware application are presented. We compare our system with respect to previous ones and demonstrate that UbiqBIOPARC is an example of how innovative context-aware applications can be built with the aid of GPS and compass features. Several real experiments have been carried out in order to evaluate performance and system behavior, and numerical results demonstrate the practicality offered by our application, while providing a quite reasonable performance in terms of delay, usability, and energy efficiency.

Key Words: iPhone SDK, mobile devices, ubiquitous computing, context-aware systems.

I. INTRODUCTION

The term ”ubiquitous computing” refers to making many computing devices available throughout the physical environment, while making them effectively invisible to the user [1].Thanks to the advances in the devices' processing power, extended battery life, and the proliferation of mobile computing devices, the realization of ubiquitous computing has become more apparent. In context-aware computing, applications may change or adapt their functions, the way information is presented, and the user interfaces depending on the context, the characteristics, and the profile of the client that is using the system. This way the information offered to the user may change depending on location, orientation, or even on his personal profile, without the user's intervention. Nowadays, context-aware applications exploit mobile wireless communication technologies to interconnect several computing devices, providing intelligent environments.

Tourism and interactive guides are one of the business application areas for the development of context-aware applications. These kinds of applications are able to provide users with personalized visits to places such as zoo thematic parks. The idea of using personal guides in museum, thematic parks, and historical sites as potential applications for pervasive computing is not new. Other researchers have chosen this application area to demonstrate the practicality of implementing context-aware applications. A lot of research has also been carried out in this area. Some examples of previous prototypes are briefly presented in Section 2.

In this paper, we present a novel application called UbiqBIOPARC, that provides context-awareinformation to zoo visitors using smartphones. In particular, iPhones are the devices chosen to develop this system. The application provides users with information about which group of animals they are viewing based on their current GPS location, and the orientation given by the digital compass. Data is classified in such a way that users are presented with information adapted to their level of knowledge (i.e., child, adult, student, advanced, expert) and in the language they choose (Spanish, Valencian, and English, for the moment). Furthermore, other options such

Manuscript Received August 02, 2014; Revised September 10, 2014; Accepted September 25, 2014. (ID No. JMIS-2014-0003)
Corresponding Author (*): Juan-Carlos Cano, Computer Engineering Department Universitat Politècnica de València, Camino de Vera S/N, 46022 Valencia, Spain. +34 963877000, jucano@disca.upv.es.
[1]Computer Engineering Department (DISCA),Universitat Politècnica de València, Valencia, Spain.
jvsorribes@hotmail.com, jucano@disca.upv.es, calafate@disca.upv.es, pmanzoni@disca.upv.es

as a Google Maps interface are also available in order to allow users to be aware of the position of the nearest animal, or to search for the animal they are interested in. To achieve the information management requirements, the application offers two main functionalities: the local option uses a database integrated in the device, whereas the remote option offers updated data from a remote database server via 3G technology. Client code has been written to provide queries to both databases. This prototype system allows us not only to confirm the correct behavior of the designed application, but also to obtain experimental data to evaluate how well GPS and 3G technologies are suited for context-aware systems.

Although other similar systems to UbiqBioparc have been presented before, UbiqBioparc includes some fairly new features, such as orientation given by a compass that enables users to know which group of animals they are physically looking at, display of the nearest animal through Google (or custom) map options, or even search for the location of the animal they are interested in and the distance to it. The system has been implemented and evaluated in the facilities of BioParc Valencia,http://www.bioparcvalencia.es. We compare our system (See Figure 1) with respect to previous works and demonstrate the novelty and unique characteristics of our system.

The rest of this paper is organized as follows: Section II reviews the state of the art of context aware computing. Section III describes the overall system architecture. Section IV presents the system functionality, and Section V includes evaluation results and discussion. Finally, in Section VI, some concluding remarks are made.

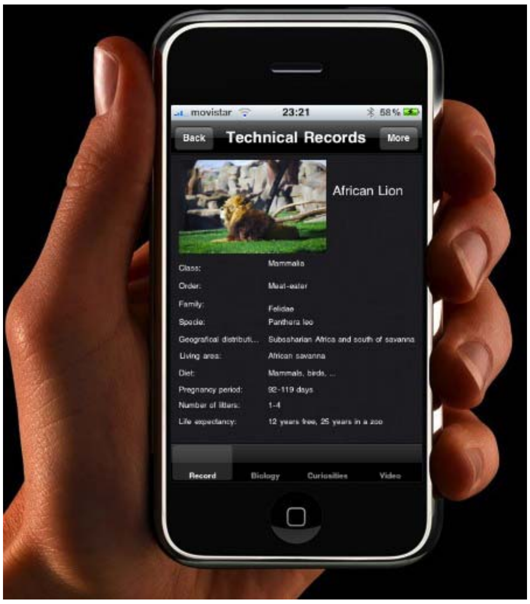

Fig. 1. UbiqBioparc application running in a iPhone.

II. RELATED WORK

The area of smart spaces and context aware systems has become one of the key engines in pervasive computing, where many different ideas and system have been proposed in the last decade. CoolTown [2] embodies web technologies into the physical environment allowing users to visualize services related to physical objects of interest. In particular, Cooltown Museum uses PDAs that can receive URLs associated with books, calendars and posters for sale, and then linking those URLs into web information about these items. These URLs will give them book reviews, available inventory of calendars, and other colors of posters. Small infrared transceivers, called beacons, are located close to pictures or sculptures, and used to broadcast URLs. Using the PDA’s web browser, visitors can read or hear about the artist, the work and related art works in the museum. The museum book-store’s web portal provides services to assist in buying books, including a service to order books that are not available. These scenarios combine local wireless communications for location-specific addressing with wider range wireless Internet access to obtain the addressed web pages. Thus, the PDAs must have wireless connectivity (e.g., IEEE 802.11), but no location-sensing or tracking technologies are needed to create a location-aware application. To track the users’ presence at a location, active badges are available to be worn by individuals and objects. So, a drawback of this system is the need for beacons, RFID tags, or barcodes associated to the objects of interest.

Websign [3] is another system which offers tourist guide services. By implementing augmented reality, the system allows users to visualize services related to physical objects of interest. The websign system provides the infrastructure not only for detecting websigns, but also for creating and deploying them. Websigns are hyperlinks from physical points in space and time to web resources. Users must carry internet-enabled wireless mobile devices such as PDAs or smart phones equipped with the client software, and positioning systems such as GPS and a digital compass. The websign-enabled mobile device downloads and stores XML descriptions of websigns in a wide area, consisting of locations of websigns, their associated URLs, and other control information. Users can point their mobile devices towards physical objects of interest to observe the real world superimposed with the virtual objects. To be linked to web resources associated with virtual beacons, users are required to point their mobile device and explicitly select one virtual beacon. The advantage of this system is that it uses GPS and compass, meaning that users just have to point to the object of interest to obtain information from it. A disadvantage is that only red spheres are displayed in the device screen and no information about them is displayed until the user selects the virtual object.

There is also a system, called Rememberer[4], that allows visitors of a museum to record their visits; each record, which can be consulted during or after the user's visit, consists of a set of web pages with multimedia data and typed notes about the visited exhibits, describing the visit; the location is determined using infra-red technology and RFID sensors. The user can register exhibits during their visit, consisting of a set of web pages, and a physical artifact that reminds the user of the visit and contains a pointer (URL) to the visit record.

Moreover, a context-aware system based on Bluetooth and Java is proposed in [5]. It provides context-aware information to museum visitors, using mobile phones, PDAs and laptops. The system provides precise information to visitors about what they are viewing, at their level of knowledge, and in their native language. It also provides a graphical user interface (GUI) adapted to their device. The application will also help reduce costs incurred in guiding their museum visitors, and in keeping track of what the preferred pieces of art are, and so on. It integrates multiple types of devices including Bluetooth, LAN and WIFI technologies; however, no orientation measures are taken into account. Furthermore, since Bluetooth just works at a distance of up to 10 meters, the user will not see anything in his device's screen when he is far from the pieces of art.

A news delivery system, described in [6] uses a set of wireless multimedia applications under the iPhone's platform, enabling context-awareness. Location and sensor awareness is provided at the client side. The web server is based on Linux, Apache, PostgreSQL, PHP, and it determines the location-awareness of the device (GPS is not used). A disadvantage is that context- awareness is reduced to location and sensor proximity, being important to extend it to take into accounts the users' profile and preferences.

Furthermore, a context-aware medical content adaptation is presented in [7]. It provides telemedicine services, based on context awareness, using sensors to determine the status of the patient monitored through a medical network. The main advantage of the system is that all the process is transparent to the patient and monitoring is carried out efficiently; however the system is difficult to set up. An innovative semantic web based tourist guide application is presented in [8]. It features a semantic web rule reasoning engine that enables visitor-oriented services to identify relevant sources of contextual information and to enforce user-specified privacy preferences about what information they are willing to share with others. It has been deployed in the National Museum of Natural Science of Taiwan. It captures and exploits knowledge about the museum's exhibits, its layout, as well as contextual information about visitors and relevant privacy preferences. A PDA asks visitors to enter their interests and preferences. Based on this information, the guide builds an initial tour, giving directions to the visitor, and being adjusted, as needed, as the tour progresses, suggesting additional exhibits or showing nearby restaurants. The main disadvantage of this application stands in the fact that maps navigation is not supplied.

Finally, in [9] a context-aware city guide with multi-modal interfaces is described. It is based on a Compaq iPAQ Pocket PC that allows tourists to receive information about city attractions. This system shows that GPS location is not accurate, giving positioning errors of about 50 meters in some occasions, and there are zones where GPS completely fails to give the right position.

Smart-phones manufacturers are also looking into location temporary applications [10], that is, companies are currently looking for patents (proposals) venturing in applications that automatically appear on the iPhone OS when you come in the proximity of certain locations, providing multiple options and information. In addition, when the user is far from the given location, the application is automatically deleted. For instance, a user can stand outside a restaurant and view the full menu on the device, make orders electronically, consult the waiting time, and so on. A summary of additional related applications can be found in [11].

Table I presents a qualitative comparison of different context aware systems including our UbiqBIOPARC system, which we now proceed to present and evaluate.

III. SYSTEM ARCHITECTURE

The overall network architecture is based on the combination of a main management server in charge of the corporative database, a wired network, and a WIFI backbone with mesh technology. The wireless network, based on 802.11 standard, consists of a set of access points which allow users to connect to the server. On the other hand, the client devices (GPS enabled smart mobile phones) use WIFI and 3G technology to connect to the main server, whereas the wired network is based on a Ring single-mode Optical Fiber at 1 Gbps. Furthermore, there is GPS coverage throughout the park. Each smart phone has a GPS chip and a compass, whose measures will be taken into account when developing the application.

Figure 2 shows the system architecture representation of UbiqBIOPARC.

Table 1.Qualitativecomparison of contextawaresystems.

	Websign	**Rememberer**	**UbiqMuseum**	**News**	**Semantic**	**Multi-modal**	**UbiqBioparc**
Native application	NO	NO	**YES**	**YES**	NO	NO	**YES**
3G	NO	NO	NO	**YES**	NO	NO	**YES**
Multimedia	NO	**YES**	**YES**	**YES**	NO	**YES**	**YES**
Smart Phones	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**
Infrarred	**YES**	**YES**	NO	NO	NO	NO	NO
RFID	**YES**	**YES**	NO	NO	**YES**	NO	NO
Bluetooth	NO	NO	**YES**	NO	NO	NO	NO
GPS	**YES**	NO	NO	NO	NO	**YES**	**YES**
Compass	**YES**	NO	NO	NO	NO	NO	**YES**
Track presence	**YES**	**YES**	**YES**	NO	**YES**	**YES**	**YES**
Augmented reality	**YES**	NO	NO	NO	NO	NO	**YES**
Knowledge based info	NO	NO	**YES**	NO	**YES**	NO	**YES**
Multi languajes	NO	NO	**YES**	**YES**	NO	NO	**YES**
User profile	**YES**	**YES**	**YES**	**YES**	**YES**	NO	**YES**
GUI adapted	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**
Keep track of objects	**YES**	**YES**	**YES**	NO	**YES**	**YES**	**YES**
Multi-modal interfaces	NO	NO	NO	NO	**YES**	**YES**	NO
Maps support	NO	NO	NO	NO	NO	**YES**	**YES**
Search and location	NO	NO	NO	NO	**YES**	NO	**YES**
Local/Remote modes	NO	NO	NO	**YES**	NO	NO	**YES**

The system is composed of (a) clients, using the iPhone device, (b) a server located in the park, which contains the remote database, and (c) an interface with clients based on ASP. The system is designed in such a way that there can be multiple clients connected to the server at a particular moment. The client side deals with GPS measurements (location), compass readings (orientation), and a local database integrated into the device, used by the native application. For information management, the application communicates with both a remote and a local database. The remote database can be updated whenever the administrator desires, and its information will be available wherever the user is located. The local database always contains the same information, and this information will always be available, even if the client device wireless connection is lost. Figure 3 shows the system client-server architecture, which allows automatically retrieving information from the remote database storing it in the local database, allowing local database management.

1. Central data server

The central management server's main function is to attend requests from the clients and offer information to them using an SQL Server database, although other database implementations are possible, as this implementation is transparent to the client side. This server must be running all the time to attend client queries.

2. Database

The database includes all the needed information about the park's points of interest (i.e. animals, plants, and landscapes), that the application will manage. It consists of 17 tables containing multimedia information about points of interest, technical records, biology, curiosities, video, audio, photos, GPS points, park services, profiles, areas, language, and types of video and audio. In addition, this database has been designed in such a way that it is compatible with other operating systems applications, such as Android, and Windows Mobile 7.

3. Database server interface with clients

The database server interface with clients is composed of an ASP script. When the remote mode is selected and information is required, the client native application sends SQL sentences (requests) to the server, including the table name and the SQL sentence as parameters. Then, the ASP script connects to the database, executes the SQL sentence, and converts the resulting information to an XML file format, sending it back to the application. Once the XML file reaches the client, it is parsed by the application's XML parser in order to extract the information and show it to the user. When the local mode is chosen, the native application directly uses the information from the local database, whose data about points of interest have a similar structure to that used in the remote database.

4. Client device

According to its functionality, we may encounter several modes within the application, i.e. local, remote, GPS searcher, nearest animal, and context-aware modes. Thus, information can be accessed in 3 possible ways, which have been implemented. (i) The user can manually search for the multimedia information he desires. This utility corresponds to local and remote modes, and it is an easy process for the user, but the user must do a great effort on looking for information.

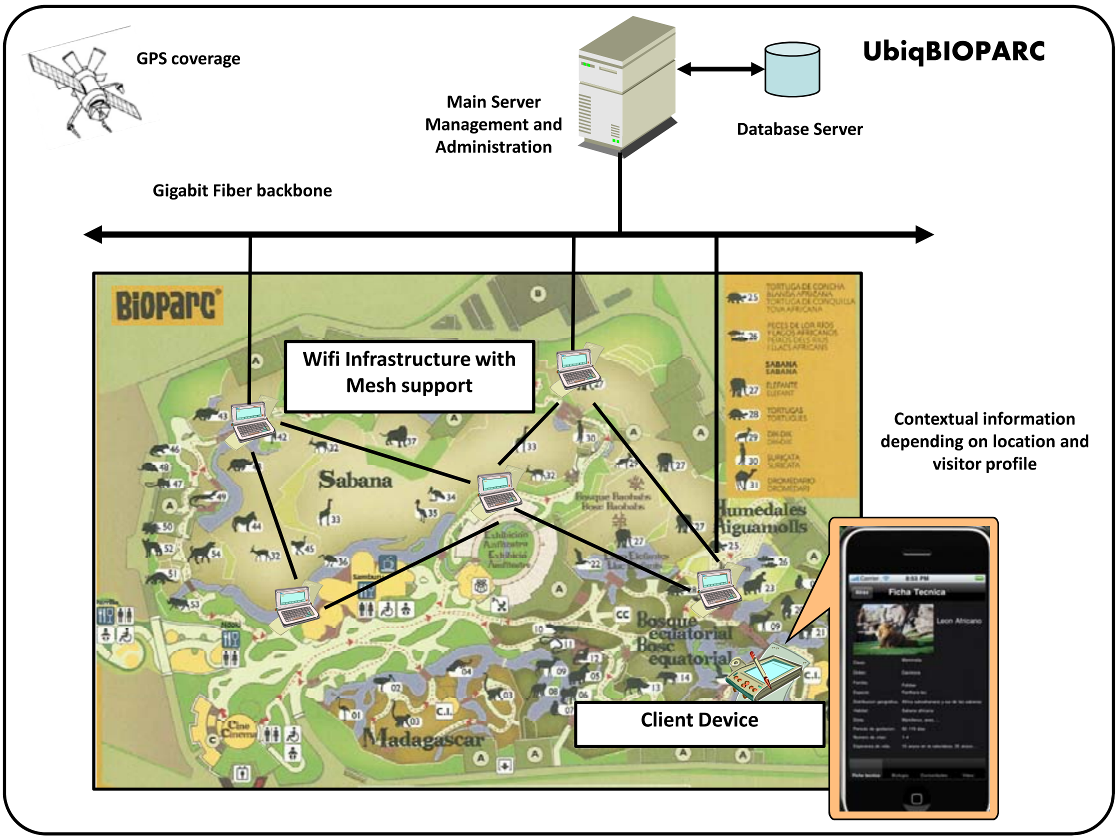

Fig. 2. UbiqBIOPARC System Architecture.

(ii) Map-based searching, which makes use of the GPS for searching animals within the map. It requires less effort from the user but this process can still be improved. (iii) The user just selects his preferences on language, level of knowledge, and context-awareness. After that, the application will automatically display accurate information without the user's intervention that will vary depending on user's location and orientation. As a conclusion, this option provides the user with appropriate information and simplifies the searching process by minimizing the user effort on searching for information.

5. Additional remarks

Experimental results demonstrate that the GPS measures given by the device are not always accurate, mostly due to errors produced by the GPS chip reading, introducing an error margin of more than 40 or 50 meters in some occasions. We also noticed that, as time goes by using the application, the measurement accuracy gets better and better. There are different accuracy options in iPhone GPS location; in particular, 1 km, 100 meters, 10 meters and ”best”. However, even the most precise option leads us to reading errors of a few meters.

Furthermore, GPS managements must be carried out so that power consumption is minimized. Taking GPS and orientation measurements continuously will result in a great accuracy, although consuming a lot of power. An alternative is to take measurements when the user has moved several meters, providing not so good accuracy but saving a lot of power.

In this paper we present a method that consists in obtaining new measurements when the user has moved 5 meters or changed his orientation by at least 45 degrees, offering good results. The rest of the time the GPS interface is asleep, so that the power consumption is minimized. In Section 5, the power consumption will be further analyzed.

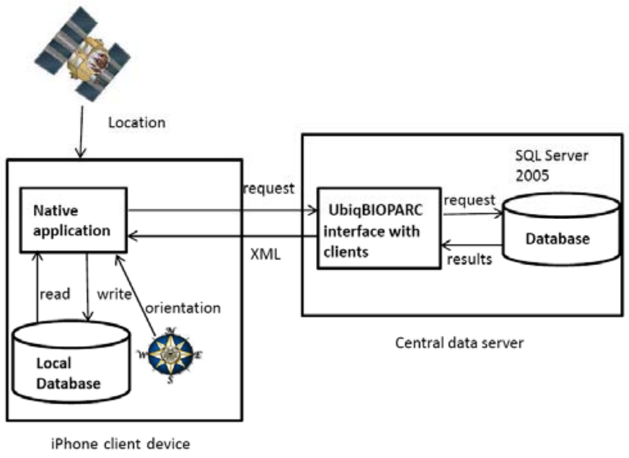

Fig. 3. The system client-server architecture.

IV. SYSTEM FUNCTIONALITY

The application was developed using iPhone SDK 3.1. In particular, the context-aware module uses the CoreLocation framework and the CLLocationManager class.

1. Local mode versus remote mode

The first step required to the user is to set the language (Spanish, English, Valencian), the level of knowledge (Child, Adult, Student, Advanced, and Expert) and to choose the WIFI OFF option, which allows visitors to select the local mode. The information handled by this mode is stored in an SQL database, which has been selected due to its flexibility and ability to store huge amounts of information. Next, users must choose the area they are particularly keen on, resulting in a list of animals, plants, and landscapes. After selecting one of them, multimedia information is displayed. Depending on the user's profile and the language, the information will vary. In addition, these two parameters may be changed whenever the user desires. By selecting the Wifi on option in our application, completely up-to-date information is obtained from the server using the Wifi interface of the client device. In this case, each screen of the application has exactly the same appearance as in the local mode.

Note that the remote mode permits using the application far from the database server's location (i.e., it corresponds to a client using the application at home, before or after visiting the park).

2. GPS location and Google Maps

Under this functionality, GPS is used to locate information on a map. In a first step, the user presses a search button, which leads him to a list of animals stored in the local database, a contextual keyboard, and a search field. In addition to the option of selecting the animal from this list, the user can introduce the animal he wants to search. As he introduces characters, the list is filtered so that only those animals whose name begins with the letters he has introduced appear.

For this purpose, several tables are used to store GPS points (basically their latitude and longitude coordinates) in the local database. These GPS points are taken around each area, that is, surrounding the places where the animals are physically located.

To summarize, the GPS search utility reads the GPS points that correspond to the animal the user wants to look for from the database, and it calculates the minimum distance from the user's device to each of these points. At the end of this process, a pin is placed in the map, showing the animal's name and the distance in meters from the user to it. By means of tapping the arrow button next to the name, multimedia information selected according to the user profile is displayed.

Fig. 4. GPS searcher interfaces.

Moreover, a nearest animal utility has also been developed to tell the user which is the position for the closest group of animals, and the distance to it, dealing again with the GPS points stored in the database, and providing multimedia information about the animals.

3. Context-aware mode

In the normal case, the user should be able to wander around the park and the application, when detecting proximity to an area, should display the list of animals of the area the user is physically viewing at that particular moment without requiring the user's intervention. After that, when an animal is selected, contextual multimedia information is shown. Once more, this multimedia information will vary depending on the user's profile and the selected language in the configuration process, which may be changed at any time going backwards to the corresponding client interface.

Fig. 5. Nearest animal interfaces.

To pinpoint the correct animal, two measurements have to be combined and taken into account, that is, GPS locations and compass readings.

Two different approaches have been proposed to implement the context-aware functionality.

- The first approach differentiates the scenario where the user is immersed in two possible types of zones: "two-area" zones, which have animals at both sides of the path, and "one-area" zones, which have animals at just one side. Figures 6 and 7 illustrate both types of zones.GPS points are taken from the border of both sides of the path in "two-area" zones, while the GPS points are taken from just one side in "one- area" zones. Therefore, the main advantage is that the number of GPS points, taken during the deployment in the park, will be lower in comparison to other approaches, saving memory in the database. However, the most important drawback is that it is not 100% reliable, as its calculations depend on the user's GPS position in "one-area" cases (which is not 100% accurate due to iPhone GPS positioning errors). So, a second approach must be used, which is almost 100% accurate in any case.
- The second approach considers the same model for "two-area" zones as in the first approach (see Figure 6), whereas "one-area" zones can be seen as having "something" at one side of the path and "nothing" at the other side (indicating that there are no animals at that side). So our new approach always uses a "two-area" algorithm, particularizing those "one-area" places with a 0 mark. Its main drawback is that more GPS points have to be considered, making the process of obtaining and introducing them into the database more time consuming. Additionally, it requires more memory to store the GPS points of animals into the database.

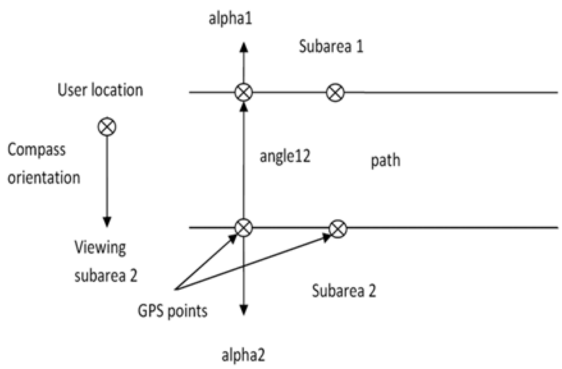

Fig. 6. Angles of vision in Two-area zones..

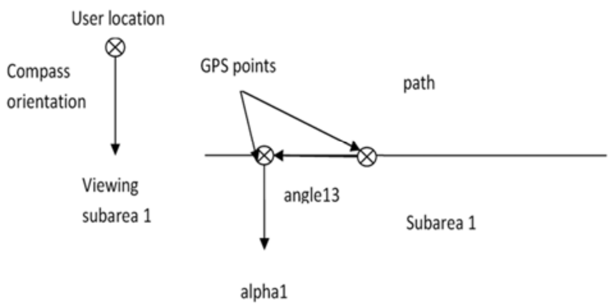

Fig. 7. Angles of vision in One-area zones.

Algorithm 1 Discover the area the user is viewing

{Find out subarea 1}
mindistance = 1000000
for all GPS points in the database **do**
Calculate d1 distance from user to this GPS point
if d1 <mindistance**then**
mindistance =d1
New minimum point p1
New subarea sub1
end if
end for
{ Find out subarea 2}
mindistance = 1000000
for all GPS points in the database except those from sub1 **do**
Calculate d2 distance from user to this GPS point
if d2 <mindistance**then**
mindistance =d2
New minimum point p2
New subarea sub2
end if
end for
{Find out the angles of vision}
Calculate angle12 (between p1 and p2)
alpha1 =angle12
alpha2 =angle12 + 180
{Discover the area the user is viewing}
if*alphauser*>=alpha1-60 and *alphauser*<=alpha1+60 **then**
area =sub1
else **if***alphauser*>=alpha2-60 and *alphauser*<=alpha2+60 **then**
area =sub2
else
area = 0
end if

For the development of both approaches, we developed the code shown in Algorithm 1, which is able to discover the group of animals the user is viewing. This process requires discovering the areas at both sides, then finding out the angles of vision to those areas and comparing user's orientation (from the compass) to those angles of vision. As a result, the area the user is viewing is obtained. For further details, see Algorithm 1.These two proposals will be thoroughly evaluated and compared in Section 5.

V. PERFORMANCE EVALUATION

Several experiments have been carried out in order to evaluate the developed application's performance. In particular, our experiments focused on evaluating the correct functionality of the application, the response time, and the power consumption. Additionally, we are interested on finding out how well the combination of GPS localization and compass orientation supports our context-aware applications.

1. Time Response Evaluation

First, we evaluate the time the user has to wait for the different views to appear after selecting the corresponding utility in local mode (without WIFIconnection) and remote mode, comparing both modes. Results are shown in Fig.8.

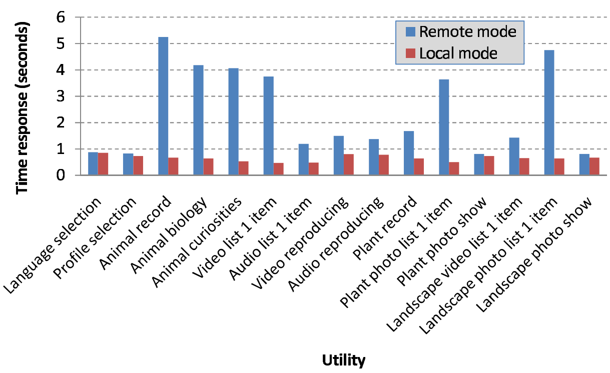

Fig. 8. Time response comparison local vs. remote mode, for the different views.

We can conclude that, in local mode, users must wait at most 1 second to view any information they are looking for. The experiments dealt with all kinds of information, including animal technical records, biology, curiosities, video, audio, plant technical records, photo viewing, and landscape data. Results prove that the user will have to wait less than 5.5 seconds to view any information in remote mode; in particular, remote information will take on average 2.5 seconds to appear, and the technical record introduces the highest time response, with 5.25 seconds. As a conclusion, the remote mode time response is always higher than local mode's throughout all the possible utilities. Note that, when we are using the application from outside the park, we need a data connection (e.g., UMTS) for the remote mode to be available.

Next, the local and remote data mode time responses required in displaying lists of points of interest are evaluated. For this purpose, 10-item list of either animals, plants, and landscapes have been considered. The results are shown in Fig. 9.

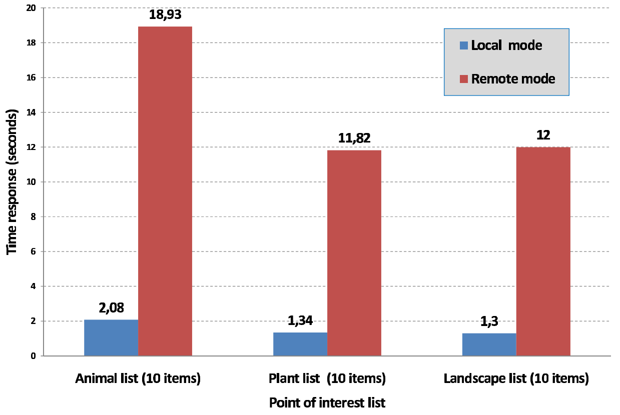

Fig. 9. Time response for different lists of items (animal, plant and landscape) in local and remote modes.

From the chart we can conclude that 10-item lists will take about 2 seconds or less to appear in local mode, whereas they will take up to 18.93 seconds in remote mode. This value is much higher than for the local mode due to the wireless communication's delay.

Moreover, experiments have been carried out to evaluate the remote mode time response when storing images in the server and storing them locally in the device. To achieve this goal, we measure the time it takes for 50 animal lists with different image sizes to be loaded. The results can be observed in Fig. 10.

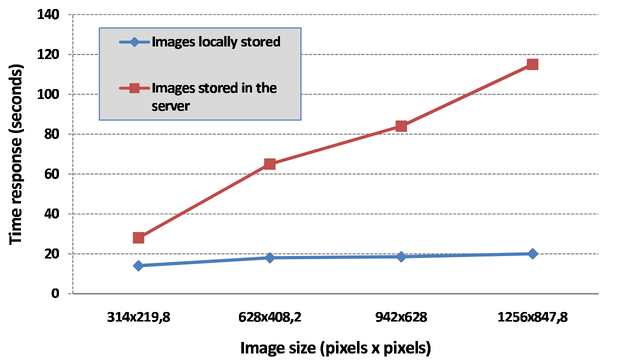

Fig. 10. Remote mode time response. Images locally stored vs. images stored in the server. 50 animal lists using 3G connectivity.

From the chart we can observe that performance is always better for locally stored images mode. Although its performance is almost the same for small sizes, differences become quite significant when the image size increases. Bases on this behavior we should decide the best trade-off between storage requirements and response time. Images stored in the server may save memory in the device, but they introduce high delays. In particular, delays higher than 25 seconds could be considered prohibitive if the user requires a low time response. However, for 1256x847 pixels images, each file is 180.5 KB size. If we consider a 50 animal list, the amount of memory needed is 9025 KB, that is, 8.81 MB. So, at the maximum image size considered, the memory required to store this list is about 8.81 MB. In this case, the photos can be properly seen although the user has to wait up to 115 seconds, which is not appropriate since it will cause users annoyance. Furthermore, if more than 50 animals are considered for in-server mode, this behavior could be even worse, and the required memory would also increase. In conclusion, depending on the number of animals considered, the memory and time response requirements, we should choose between the locally stored images mode and the in-server stored images mode. In this sense, the required memory to store lists of images can be easily determined by multiplying the memory size for one image and the number of animals. Figure 11 has been obtained for a good image resolution of 942x628 pixels and compares the remote mode time response for different number of animals in both cases: in-server stored images mode and locally stored images mode.

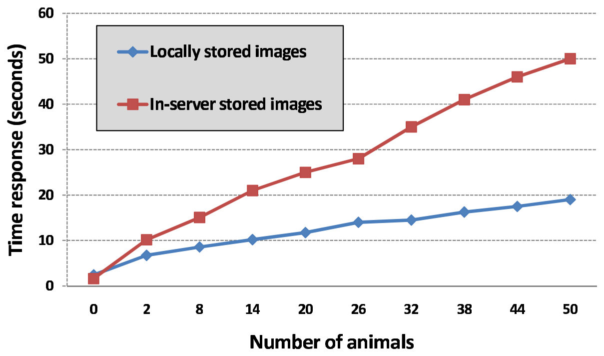

Fig. 11. Remote mode time response, for different number of animals.

The chart shows that performance is always better for locally stored images than for in-server stored images. Although the performance is almost the same for 2 animals lists, it differs more and more as the number of animal grows. In 50-animal lists, the time response is of about 20 seconds if the images are locally stored, increasing up to 50 seconds if the images are stored in the server. We also evaluated the time response delay when focusing on the Animal Search mode. In this particular case, we will measure the time required from the instant when the user selects the desired animal to the time the pin is displayed in the map. Results regarding these experiments show that the user will have to wait for the application to locate the animal in the map for about 1.36 seconds on average, and he would have to wait for the response at most 1.68 seconds, which is a very reasonable value.

Finally, the context-aware time response has also been evaluated by carrying out several experiments in such a way that the measurements are taken when turning around from an area with animals to another area at both sides of the user's path. According to Fig. 12, the user will have to wait at most 5 seconds for the list of animals to change to the new area he is viewing. As previously commented, two approaches associated with the context-aware mode have been proposed. The first approach is not 100% accurate due to GPS positioning errors. In particular, tests have been carried out to conclude that this approach has an accuracy of around 80%, whereas the second approach is almost 100% accurate and should be employed for high accuracy requirements. The GPS location given by the GPS chip may differ by about 40 or 50 meters from its real location, which happens when our application uses the GPS chip for the first time. However, as time goes by, the accuracy improves, reaching much lower error levels (about two meters).

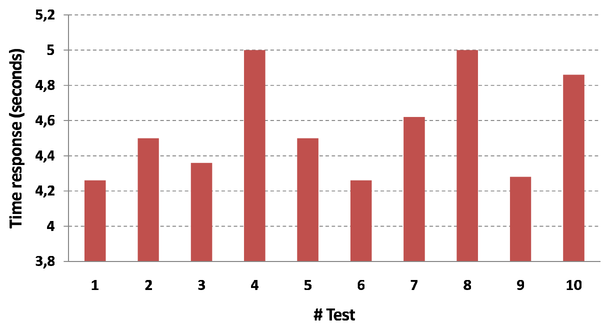

Fig. 12. Context switching time response, changing orientation from an area with animals to another area with animals.

2. Power consumption Evaluation

Finally, power consumption is an important parameter that we should evaluate. This parameter will show us how long the battery will last, that is, the amount of time that the visitor will be able to use the application before having to recharge the battery. In this study, we focused on the different common functionalities such as animal searcher, nearest animal, context-aware, local textual, local video, and local audio.

From the results shown in Figure 13, we observe that the user will be able to use the application for at least 3.33 hours in any case before having to recharge the battery, a worst case result when continuously using the nearest animal utility. On average, the user will be able to use this application for about 5 hours. On the other hand, when the user looks up local textual information and photos, he will be able to use the terminal for 6.58 hours before having to recharge the battery.

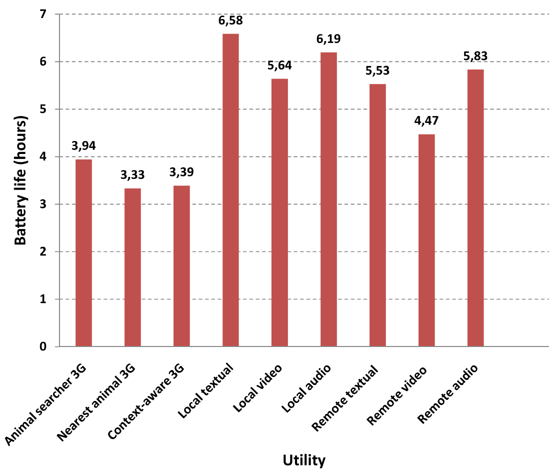

Figure 13. Battery life under different conditions.

We concluded that GPS and compass utilities will suppose a higher consumption than others, and remote options will cause more power consumption than local options. Also, the video option will suppose a higher consumption than textual and audio options. To summarize, if the visitor uses all the application utilities in a balanced way, he will be able to use the terminal for about 5 hours without having to recharge its battery, which is quite reasonable.

VI. CONCLUSIONS

In this paper we demonstrate that GPS and 3G might be a candidate wireless networking technology to support context-aware applications. UbiqBIOPARC was presented as an experimental GPS, compass, and 3G-based context-aware application developed using the iPhone SDK, and designed with the goal of providing zoo visitors with information about what they are viewing adapted to their level of knowledge and in the language they prefer. GPS location and compass readings have been managed to provide map-based utilities on animal searching and context-awareness, which shows the list of animals the user is viewing, and changes it as the user's location or orientation change without the user's intervention.

Performance evaluation of the application has been carried out focusing on time responses, and power consumption. The impact on time response was evaluated in local, remote, nearest animal, animal searcher, and context-aware modes. We observed that the application offered a local mode time response of about 1 second in any case. As for the remote mode, reasonable values are also obtained. Two approaches have been proposed in order to implement context- awareness. While one of them has 80% accuracy depending on the GPS positioning errors introduced, the other is almost 100% accurate.

Power consumption within all the modes has also been evaluated, and experiments prove that the user will be able to use the application on average, for about 5 hours, and so the battery will last for enough time to visit the park.

Further research work should be done to improve the remote mode time response, and to increase even more the battery life time; overall, the practicality of building a context-aware system has been demonstrated.

Acknowledgement

This work was partially supported by the Ministerio de Ciencia e Innovación, Spain, under Grant TIN2011-27543- C03-01.

References

[1] M. Weiser, *The Computer for the 21st Century*. Scientific American 256, pp. 94-104, 1991.

[2] T. Kindberg, "People, Places, Things: Web presence for the real world," Technical Report HPL-2000- 16. Internet and Mobile System Laboratory, Laboratories Palo Alto, 2000.

[3] S. Pradhan, C. Brignone, J. Cui, A. McReynolds, and M. Smith, "Websign: Hyperlinks from A Physical Location to the Web," Technical Report HP Laboratories. http://www.cooltown.hp.com/.

[4] M. Fleck, Ma. Frid, T. Kindberg, E. Obrien-Strain, and M. R. Rajani, "Rememberer: A Tool for Capturing Museum Visits," *Proceedings of the Ubiquitous Computing International Conference*, Ubicomp, 2002.

[5] J. C. Cano, P. Manzoni, C. K. Toh, "UbiqMuseum: A Bluetooth and Java Based Context-Aware System for Ubiquitous Computing," *Wireless Personal Communications*, Vol. 38, No. 2, pp.187-202, July 2006.

[6] J. Rodrigues, M. Oliveira, "BinodVaidya- New Trends on Ubiquitous Mobile Multimedia Applications," *Eurasip Journal on Wireless Communications and Networking Volume* 2010, doi:10.1155/2010/689517.

[7] C. Doukas, IliasMaglogiannis, K. Karpouzis, "Context-Aware Medical Content Adaptation through Semantic Representation and Rules Evaluation," *Eurasip Journal on Wireless Communications and Networking*, Volume 2008, 2008.

[8] S. C. Chou, W. T. Hsieh, F. L. Gandon, "Norman M Sadeh, Semantic Web Technologies for Context-Aware Museum Tour Guide Applications," *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, 2005.

[9] C. Borntrager, K. Cheverst, N. Davies, A. Dix, A. Friday, and J. Seitz, "Experiments with Multi-Modal Interfaces in a Context-Aware City Guide," *Springer* , 2003.

[10] Apple, Apple looking into location-temporary apps. http://macapper.com/2010/05/18/apple-looking-into-location-temporary-apps/ (May 2010).

[11] DimitriosRaptis, N. Tselios, NikolaosAvouris, "Context-based on design of mobile applications for museums: A survey of existing practices," *Proceedings of the 7th international conference on Human computer interaction with mobile devices and services*, 2005.

[12] K. Luyten and K. Coninx, "Imogl: Take Control over a Context-Aware Electronic Mobile Guide for Museums," *Workshop on HCI in Mobile Guides, 6th International Conference on Human Computer Interaction with Mobile Devices and Services*, Glasgow 2004.

[13] K. Cheverst, N. Davies, K. Mitchell, A. Friday, C. Efstratiou, "Developing a context-aware electronic tourist guide: some issues and experiences", *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp.17-24, 2000.

[14] K. Cheverst, N. Davies, K. Mitchell, A. Friday, "Experiences of Developing and Deploying a context-aware tourist guide:" the GUIDE project, 2000.

[15] J. K. Zao, S. C. Fan, M. H. Wen, C. T. Hsu, C. Hung, S. Hsu, and M. C. Chuang, "Activity-Oriented Design of Health Pal: A Smart Phone for Elders' Healthcare Support," *Eurasip Journal on Wireless Communications and Networking Volume*

2008, 2008, doi:10.1155/2008/582194.
[16] Objective-C: Use NSXMLParser. http://www.radupoenaru.com/objective-cuse-nsxmlparser/
[17] iPhone Tutorial: Adding a search bar in TableView. http://blog.webscale.co.in/?p=228
[18] YouTube.Xcode iPhone SDK Tutorial.Playing a Video File With Controls http://www.youtube.com/watch?v=9hL5jREBwu4
[19] Tutorial Request: Core Location. http://www.iphonedevsdk.com/forum/tutorial-requests/2343-tutorial-requestcore-location.html.
[20] Compass Programming Guide iPhone Dev SDK Forum. http://www.iphonedevsdk.com/forum/iphone-sdk-development/21553-compass-programming-guide.html.

Authors

Jose Vicente Sorribes earned a Computer Engineering degree from UJI and an MSc in Computer Engineering from UPV in 2007 and 2010 respectively. Since 2006, he has worked as research assistant, and from 2009-2010 he also worked as assistant professor.

Juan-Carlos Cano is a full professor in the Department of Computer Engineering at the Polytechnic University of Valencia (UPV) in Spain. He earned an MSc and a Ph.D. in Computer Science from the UPV in 1994 and 2002 respectively. From 1995-1997 he worked as a programming analyst at IBM's manufacturing division in Valencia. His current research interests include Vehicular Networks, Mobile Ad Hoc Networks, and Pervasive Computing.

Carlos T. Calafate is an associate professor in the Department of Computer Engineering at the Polytechnic University of Valencia (UPV) in Spain. He graduated with honors in Electrical and Computer Engineering at the University of Oporto (Portugal) in 2001. He received his Ph.D. degree in Computer Engineering from the Technical University of Valencia in 2006, where he has worked since 2005. He is a member of the Computer Networks research group (GRC). His research interests include mobile and pervasive computing, security and QoS on wireless networks, as well as video coding and streaming.

Pietro Manzoni is a full professor in the Department of Computer Engineering at the Polytechnic University of Valencia (UPV) in Spain. He received the MS degree in computer science from the "Universitá degli Studi" of Milan, Italy, in 1989, and the PhD degree in computer science from the Polytechnic University of Milan, Italy, in 1995. His research activity is related to wireless networks protocol design, modelling, and implementation. He is member of the IEEE.