Evolving museums into pervasive services

Miguel Ángel Martínez, Francisco Moya, Maria José Santofimia,
David Villa Alises, and Juan Carlos López

Computer Architecture and Networks Group

University of Castilla-La Mancha

Paseo de la Universidad 4, 13071 Ciudad Real (Spain)

[miguela.martinez, francisco.moya, mariajose.santofimia, david.villa, juancarlos.lopez]@uclm.es

Abstract

Museums did not change much in the last decades. Most of them are still focused on static exhibition of artwork where visitor interaction is at best constrained to the use of audio guides with static audible recordings. In our opinion they persist on an obsolete model missing the potential of the information technologies to complement and enrich the visitor experience.

This paper describes a new conception of museums, where the classic view is enhanced with an extensive support of pervasive services. Moreover, this work also presents AIMTOUR, an affordable solution built on this new conception. This system proves that technology is mature enough to offer pervasive services at a reduced cost.

1 Introduction

Nowadays, when we step into a museum we may only know how old is it from the furniture. Information technologies are missing in most of them. Even in the best case scenario we can hardly find audible guides with static recordings activated by dialing a code. Although these systemas were valid in the 80's and 90's, today it is possible to offer more and better services to improve the interaction.

It must be taken into account that the current offers of leisure activities already employ the available technology to provide services. Besides, these new technologies provides an easier and more convenient access to information. These facts make these services much more attractive than other cultural alternatives and therfore they have direct impact in medium term survivality of museums (specially for the smallest ones).

To guarantee the survival of many of these museums it is necessary to renew the museum conception and to integrate it into the current information society. To this end it seems essential to integrate the information technologies into the museum and to widen the perception held by many people about museums; adopting the approach here proposed leads to museums characterised for being interactive, customized and not only constrained to indoors exhibitions but also to large outdoors areas.

2 Related Work

There are several previous works discussing similar topics; that is, the exposed problem involves several matters such as the need for a middleware, the need for information providers, the need for location services, etc.

On one hand, there are several works in the middleware field used in pervasive computing. One of these works [1] uses web services to model the communications between museum elements. In other work, a context aware middleware for ubiquitous systems (CAMUS)[2] is proposed. Besides, in [3] a middleware for sharing sensor nodes among smart spaces called D-uMiddles is presented. In line with this work, the Mobile Gaia middleware [4] is based on the creation and connection of cluster isles. Also, there are commercial approaches, for instance in [5] they introduce Atlas, a service-oriented sensor platform that can only be programmed in Java. All this works share a common feature, none of them can be used in a heterogeneous environment (several networks, devices or languages). We use a heterogeneous middleware that can be integrated in these environments and also support several languages requiring little effort.

On the other hand, there are works that solve specific problems related to the main topic. In this regard, the work in [6] identifies various kinds of failure conditions while providing recovery mechanism to these conditions. Other works focus the location problem, such as [7],[8], [9] or [10]. In the first paper they present a location aware services scheme; in the second one, they design and implement a network agent for Bluetooth devices. The third paper contains an excellent work that provides a design and a implementation of a space service. In the last paper, the authors provides an architecture that adapts the multimedia contents to the client requirements; Despite partially solving the problem, the lack of standards make these solution very hard to combine. In our proposal we overcome this restriction by coherently adopting the use of standards. By doing so, the integration process is not only supported by almost straightforward.

Some other works have focused on the visitor point of view and do not support our museum conception. For example, the PEACH-IL project (Personal Experience with Active Cultural Heritage - IsraeL) [11] explores intra-group context-aware communication services. For this purpose, they provide visitors with specific PDAs and the visitors can make annotations, send and receive messages to other visitors, and they may also use all this information to discuss the visit.

Other works have focused on the architectural point of view and do not offer interaction methods to the visitor. Specifically, in the "The INtelligent Airport" (TINA) [12] developed a next generation advanced wired and wireless network to meet the potential requirements for future "intelligent airports".

Moreover, there are some approaches that provide solutions to the complete problem. The Bahaus-University Weimar has been working the PhoneGuide [13] since 2003. This project utilises pervasive Bluetooth tracking and computer vision techniques to provide contextual information to the visitors. Moreover, the system does not depend on servers; in fact, it is designed to be run in mobile devices. In the line with the previous works, Miyashita *et al.* presents an augmented reality techniques to provide indoor services [14]. In Kawashima *et al.* [15] they present GUPSS: a modular service-oriented architecture. GUPSS provides a specific platform to create smart spaces applications; however it is only designed for indoor systems and specific hardware (SunSPOT wireless devices). All these aproximations do not satisfies the associated problems to the new museum conception; that is to say, the museum can only be placed inside a building, this architecture be extrapolated to a really distributed museum. Our proposal can be used everywere without architectural changes, and can be integrated with other proposals.

In this paper we revise all the interaction methods that are presented in these works and are intrinsic to the museum. Moreover, we provide a more general and realistic view about museums that completes the previous ones. Furthermore, we have created the system AIMTOUR. In contrast to all these works, this system is completely distributed and multi-platform, and it is focused on providing several services to visitors and administrators, maintaining a low-cost price.

3 The new concept

Existing technologies support the implementation of the museum conception envioned here through three fundamental aspects: customization, new interaction methods and the change of the physical perception of the museum.

3.1 Customization

Visitor profiles are far from being unique and, for this reason, it is not very convenient to provide them with the same visit tour.

The museum approach proposed here adapts itself to the visitor profile, offering a customised visit depending on the visitor interests or needs.

3.2 New interaction methods

To carry out a visit customization it is necessary for the system to know the visitor. For this purpose, the museum needs to be supported on has mechanisms to learn and carry out this customization. Nowadays, it is possible to utilise several methods to achieve this aim. These methods fall into three families.

1. Explicit interaction

   This is the traditional interaction method. Through this interaction the visitor indicates the system what they prefer, and the system acts accordingly.

   This is the audio-guide model in which the visitor dials a digit and the system reproduces a message. However, this method could support different and more interesting ways of interactions. Considering that the visitor profile can be retrieved from the information the user is selected, the system could use this information to adapt the visit tour to the visitor preferences.

2. Geographical interaction

   This interaction methof supports the system to know the visitor's location at each moment in time. The position can have several precision degrees, and the system can deduce different conclusions with each of them.

   Combining this geographical information with time stamps or other information derived from others interaction methods, it is possible to improve the customised visit.

3. Audio-visual interaction

   This interaction method enables the system to see and to listen through the visitor's point of view. It can be performed in several ways such as snapshots, audio-samples, or a more sophisticated way via audio-video streaming.

3.3 Changing the physical perception of the museum

Nowadays, when we talk about museums we think of a building with a numerous artworks of several types. Despite being valid, this view is incomplete. A museum represents more than a simple building; that is, we can talk about natural museums like a natural park. We can talk about history museums like a collection of Roman ruins. We can talk about cultural museums like the Quixote route (the villages and places visited by the hidalgo knight); To summarize, a museum represents more than a simple building that stores works of art, and it can be stretched over a large geographical area.

Currently, this museum conception is fully viable, and in several cases it is essential. In order to make museums to regain a better status among leisure activities, they must include an additional value to their contents. This can be easily achieved by using new information technologies. Otherwise museums will become outdated, specially the small ones, that as a consequence of the decreasing number of visitors will fall into oblivion.

4 Test System: AIMTOUR

In order to demonstrate that this new museum conception is feasible, even under low resources circumstances, we have developed AIMTOUR. This system supports all the aforementioned interaction methods. The system provides visitors with route and information access services, along with administration services supporting them in the decision making task.

The system employs several interaction methods to customise the visit. Specifically it makes use of explicit and geographical methods. The first method obtains initial information about the visitor and detect the information requested by the visitor. On the other hand, the geographical interaction allow system to detect the visitor's position in a transparent way. Moreover, the system considers from its initial conception, the introduction of audio-visual interaction methods.

Next we enumerate the design, architecture, and implementation considerations that are necessary to tackle the development of these type of systems.

4.1 Design

In this section we explain two key issues distributed heterogeneus systems: the middleware, and the standards.

4.1.1 ZeroC Ice

The nature of the system is fully distributed. Each piece of the system are located in different devices, with different features such as mobile devices, servers, or small computers. This work advocates for ZeroC [1] to provide a middleware solution that addresses all these features.

Ice is an object-oriented middleware platform that supports many programming languages, and it is available under the GNU General Public License (GPL). It was created with the idea of overcoming the mistakes of other distributed platforms, especially CORBA.

Ice is supported on Slice (Specification Language for Ice) for decopling object interfaces from their implementation. Slice sets a contract between client and server by describing the types and object interfaces used by an application. This description is independent of the implementation language, so it does not depend on the languages used for both client or server.

Ice can create a computers grid that is remotely manageable through the IceGrid service. Moreover, it supports the application distribution from a central computer to grid nodes by means of IcePatch2; that is, we can manage the software distribution remotely. Moreover, it provides an efficient publish/subscribe service called IceStorm. The middleware offers more services, but only the most populars have been mention here.

The use of Ice determines the system's design. In all moment we must think about information interchange and the design of slice interfaces. It could seem that the use of Ice makes the system difficult, however it is simplified, because a very important system part such as communications is solved.

4.1.2 Standards

Another important aspect at design time of this type of systems is the information processing and publication. Specifically this system employs geographical information, texts, audios and videos, and it is necessary to define the way the data travels between devices. In order to address these issues, this work advocates for standardize solutions.

The future and the adaptation capability of the system are linked to the use of standards. The use of standards will ease the integration of external applications or libraries into the system; Moreover, this fact can be crucial to the system survival because it is not only a cut off system but also it is a component that other systems can use. To summarize, we can reduce the time of implementation and maximise the reuse degree, and all thanks to use of standards.

The above statements are true as long as we utilise the suitable standards. Specifically, the system tackles several matters such as location, route services, streaming, etc. For each matter we have chosen the standard or protocol more widely accepted by the research community. Next we enumerate the used standards.

1. Mobile Location Protocol (MLP)

   To locate the visitors and the museum objects it is necessary to define the location concept. For this purpose we use the MLP [16] of the Open Mobile Alliance [2].

   This standard defines geometrical shapes, the location object and some interfaces to manage the location information. Our system makes use of an implementation in slice language to manage all the system location information. So, the location information can be accessible by other systems or applications.

2. OpenGIS Location Service (OpenLS)

   OpenLS [17] is one of the OpenGIS standards and specifications, as defined by the Open Geospatial Consortium (OGC) [3].

   The Open Geospatial Consortium, Inc (OGC) is an international industry consortium of 385 companies, government agencies and universities participating in a consensus process to develop publicly available interface standards. OpenGIS Standards support inter-operable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream information technologies.

   This standard defines specific interfaces that enable companies in the Location Based Services (LBS) value chain to "hook up" and provide their pieces of applications. These applications are enabled by interfaces that implement OpenLS services such as a Directory Service, Gateway Service, Geocoder Service, Presentation (Map Portrayal) Service and others.

[1] ZeroC Ice. ZeroC Inc. http://www.zeroc.com

[2] http://www.openmobilealliance.org

[3] http://www.opengeospatial.org

Our system employs two OpenLS services; specifically, it uses the directory service and the route service. The directory service provides interfaces that must answer queries such as “Is this point present within this area?” or “Where is the point of type museum object nearest to the point where I am located?”; whereas, the route service provides interfaces that must obtain routes between points.

Our system utilises an implementation [18] of these two services using the middleware ZeroC Ice. Specifically, this implementation uses an OpenGIS database (PostGIS) to store the geographic and routing information, and this implementation provides all the interfaces the two services.

3. HTTP and RTSP

   It is necessary to provide visitors with the required resources. These resources can be texts, audios or videos. Therefore, each of these resources have to be separately analyzed.

   First, we have studied the text resources. Texts can be provided with or without streaming. If we utilise a streaming solution, this can minimize the used net bandwidth, on the other hand we can use the HTTP [19] protocol for this purpose and we can obtain advantages such as support for several file formats. This is an interesting strenght for the system, and this is the reason to choose HTTP to provide texts.

   Secondly, we have studied the audio and video resources. The variable bit-rate of this files is a limiting factor run time. This factor determines the use of streaming for audio and video resources. We have used RTSP [20] to this purpose because it is a *“de facto”* standard in this field; besides, this protocol is widely accepted by the applications that work in this field.

4.2 Architecture

Taking the design aspect into consideration, we have created a completely distributed architecture that supports all the system services.

The system has been divided into independent subsystems; Each one works independently at both logic and physic level; in this regard, we can delimit fails, independently implement the subsystems, deploy each subsystem in a specific computer, etc.

Specifically, we have divided the system into ten subsystems that are explained underneath and depicted in figure 1.

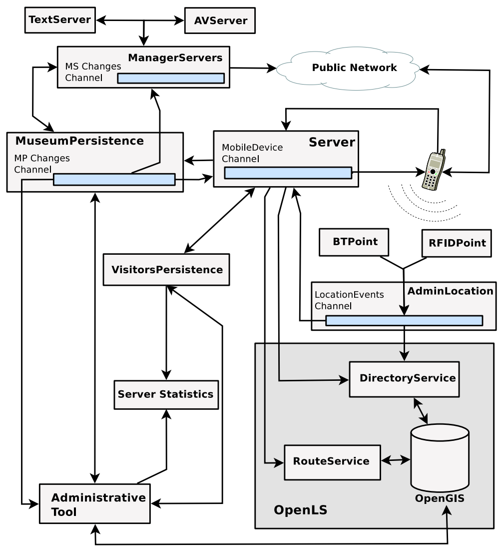

Figure 1: AIMTOUR Architecture

4.2.1 Server

This Server supports information interchange among mobile devices and the rest of the system. Mobile devices request services to the Server, and it responds to the mobile devices through an event channel or through the direct proxy to the mobile device.

This subsystem utilises some mechanism to reduce the response time; Specifically, it uses a cache to store museum data (the database can be in other computer) and several threads to manage the requests.

4.2.2 Museum Persistence

The museum data are stored in this subsystem. To store all the museum data we can use traditional databases, but we have selected the Freeze[4] technology to this end for several reasons. Specifically

[4]Freeze. ZeroC Ice manual. Pages 1443-1538

we have chosen Freeze Maps because they provide persistent support to store Ice Objects, efficient reverse lookups, and (of course) because they are completely integrated with Ice.

This subsystem allows other subsystems to be informed about the changes in the persistence. To support this feature, the subsystem implements the observer design pattern and it publishes each change in a event channel. In this regards, the interested subsystems can subscribe an specific event channel to receive the changes notifications.

4.2.3 Visitor Persistence

To customize the visitor's visit, it is necessary to store their current states; that is, the system needs to know the characteristics of the visitor (such as their language, visit time, if they are adults, childs or disabled people, etc), the visited places, the followed route, the consulted documents, etc. With a correct manipulation of this information the system can adapt the visit to the visitor.

We must take into account that there are many accesses to this information when the visits are online. To relieve this we uses the Freeze Evictor.

Freeze Evictor utilises the Evictor design pattern to separate the active objects from the inactive stored objects, enabling stable switches among objects. With this mechanism we have a subsystem that stores the visitor current state in a very efficient way.

4.2.4 Server Statistics

When the visitor exits the system their associated information turns into static information. This information represents the system use statistics and it must be stored to generate reports.

The subsystem turns the visitor object into statistic information and stores it, using Freeze Maps for this purpose. Moreover, the subsystem provides interfaces that allow to recover all this information in several ways (by date, by order, etc), favouring the statistics generation.

4.2.5 OpenLS Service

This subsystem provides a route and a directory service thanks to a OpenLS implementation [18]. This service provides support to calculate routes between points and to recover the nearest point to a given one.

4.2.6 Multimedia Server

The management of the multimedia contents can be a critical factor at run time. To overcome this issue we have the multimedia server.

This subsystem manages the texts, audios and videos in a separated way. In fact, it is possible to have several multimedia servers at the same time; by doing so, we can manage each media type with several dedicated multimedia servers. Specifically, we have a ManageServers that registers all the media servers with the media type they serve. Multimedia servers are therefore transparent to the rest of the system.

Each media type is treated with regard to its nature. The texts media have a delimit size and can be present in several formats, such as HTML or PDF. Using HTTP to serve this text eases the recovery process by resorting to web browsers. For this reason we have chosen this method to provides texts to visitors.

The audio-visual media have a different nature. These media usually have a big size, and in most situations we do not know the real size. These special features make this media type to require a streaming process to be served. For this purpose we used RTSP because it is the standard supported by the great majority of devices, both on the server and client side.

Each MediaServer is subscribed to the persistent event channel, and updates its state when a change from the persistent subsystem is received. Moreover, the MediaServers provides interfaces to request media to be played or stopped, and to notify about changes in the subsystem.

4.2.7 Mobile Device Application

The system is pointed to provide services to visitors, and this is possible via the mobile device application. This application provides a simple user interface through which the visitor interacts with the museum in a natural way. With this application the visitor can recover the contextual information provided by the museum. Moreover, he or she can ask for routes between museum points in a similar way.

For this purpose the system asks for a basic initial information such as the type of user (adult, child) or the time to spend in the visit. Next, the system registers the visitor and adapts the information to his or her context. The information is presented as a list. Each item of list contains a descriptive icon (that represents a text, video, information element, etc) and a small text description. When the visitor selects an item he or she receives the media (if it is a media element such a text or a video) or the route to reach to the point.

4.2.8 LocationAdmin

This subsystem manages the location events. This subsystem allows every location event provider to propagate them in a controlled way. To do so, the Ice implementation of the MLP standard provided by the OpenLS implementation has been used.

4.2.9 BTPoint and RFIDPoint

To provide location events we have utilised two technologies: Bluetooth and RFID. These subsystems propagate the location events in the same way; in fact, it is possible to utilise other technologies that use the same propagation method. This is possible because we utilise an implementation of the MLP standard; therefore we obtain a hierarchical and scalable location event propagation system. That is, all the system reacts in the same way when location events appear, regardless of the provider.

4.2.10 Administration Tool

The administration tool can manage the information system. That is, the tool allows to assert information about documents, museum points, to recover reports, etc.

This subsystem knows the museum persistent state, the visitor's state and the statistics, because it has proxies to all the necessary subsystems. Moreover, it is subscribed to all the necessary event channels; by doing so the system is updated at all time.

Several factors must be taken into account when designing the user interface; first, the complexity of the user interface, as it must deal with the concurrent execution of several threads. Secondly, we must consider the appearance and the quality. For this purpose we have used the Gnome Human Interface Guidelines[5].

4.3 Implementation

At implementation time, there are several aspects to be considered such as the libraries and language choice, the mobile devices and their technologies, the software tests and the design patterns; next we explain these aspects.

4.3.1 Libraries

Another important factor is the correct use of libraries. This factor can minimize the system implementation time and increase the system reliability, but only if we choose the right libraries.

There are some fields in which we use libraries; for instance we used Bluetooth and RFID and they are supported by using libraries [6,7]. Also we used graphical libraries [8,9], multimedia libraries [10,11] and server software[12,13].

4.3.2 Languages

The use of a middleware solution prevent us from having to be constrained to just one implementation language; therefore, each of the composing subsystem can be implemented in the more suitable language.

A very interesting issue to be considered is the possibility of creating a multi-platform system (or, at least, a part of it). If we take it into account we can consider languages such as Python [14] or Java [21].

The iterative implementation method benefits from the use of a prototyping language. Moreover, this language must interact with a lot of libraries and system calls. Python satisfies all these features and it has been chosen for that reason.

[5] http://library.gnome.org/devel/higbook/stable/index.html.en

[6] BlueZ. Official Linux Bluetooth protocol stack.

[7] MUSCLE. Movement For the Use of Smart Cards in a Linux Enviroment

[8] GTK. http://www.gtk.org

[9] Mapnik. http://www.mapnik.org

[10] FFMpeg. http://www.ffmpeg.org

[11] GPCA. http://gpac.sourceforge.net

[12] Darwin Streaming Server. Apple Inc. http://dss.macosforge.org

[13] minihttpd. http://acme.com/software/mini_httpd

[14] Python Software Foundation. http://www. python.org/fsf

Moreover it is necessary to store information in the system, and it is a task for Freeze; Freeze works only in C++ or Java so we need to implement the persistent subsystems in one of these languages. We have chosen Java for this purpose because it has native multi-platform support.

4.3.3 Mobile devices

Nowadays there are several mobile operating systems (Symbian, Windows Mobile, IPhone OS, Android...) so therefore, it is not feasible to develope an application of each of these systems. Since the great majority of them support J2ME [22], we have chosen this technology (also supported by the middleware) to develop the mobile application.

4.3.4 Software Tests

The most effective method to make sure that the software works properly is using software tests. By means of specific tests for each subsystem we can ensure its correctness, and by making complete tests for all the system we ensure the correct work of the whole system.

For this purpose it is very interesting, and really necessary, to automate the testing task; that is, all the system has been checked with tools [15][16] which, along with a good design, guarantees a high quality level.

4.3.5 Design Patterns

A fundamental aspect to support the system customization is the correct use of design patterns. For example, the system can supports several technologies that provide location events, and it can manage and prioritise some events above others. To solve this situation we utilise the *Factory* pattern. Another using example of design pattern can be seen in the notification of changes through *Observer* pattern.

5 Future

The open architecture of the system supports improvements and changes. The first and most immediate improvement is the creation of a GPS location agent in the mobile device. In this regard, the visitor could utilise the system in the same way they did inside the museum, and it is not necessary to modify the existent system.

Another interesting improvement could be the introduction of intelligent agents which can determine the visitor's profile with high precision. The agents can use all the information provided by system, such as location events, consulted documents, visited places, etc. Moreover, it is possible to replace the intelligence of the system by replacing the class that implements this intelligence, and this change will be transparent to the rest of the system.

It is also possible to incorporate the image management as an interaction method between the museum and the visitor. This can be implemented in several ways. In one of them, the visitor takes a photography with his or her device and the photo is sent to the server. The server receives the photography with the position in which it was taken and returns the image enhancing it with relevance information. Or, in another way, we can utilise increased reality taking again the location and the image from the device as basis.

We are currently working in all these aspects. We are working in the GPS mobile agent for Android and J2ME. Moreover, we are drawing up implantation plans to several types of museums.

6 Conclusion

In the next few years the current conception that we have about museums and oriented tourist services will complettely change. The introduction of mobile devices with Internet connection everywhere, high processing capacity, better batteries and more functionalities will cater to redesign completely the interaction methods with the environment. In this regard, we will obtain a service customization in a predictive way.

The future of museums and all of these services should go through the change of the old conceptions and the provision of new approaches according to the existing technologies.

Acknowledgements

This work has been partially funded by the European Regional Development Fund Regional Gov-

[15] Atheist. http://crysol.org/en/node/1218

[16] GNU Make[23]

ernment of Castilla-La Mancha, and Ministry of Science and Technology under grants PAI08-0234-8083(RGrid), and TEC2008-06553(DAMA).

References

[1] S. Cruz-Lara, J. Ducret, J. Faivre-Vuillin, Eing-Jieh Chang, Bai-Hsuan Chen, and Jen-Shin Hong. Using web-services for accessing and sharing multimedia resources in distributed digital museum. In *Multimedia Software Engineering, 2003. Proceedings. Fifth International Symposium on*, pages 54 – 61, dec. 2003.

[2] S.L. Kiani, M. Riaz, Sungyoung Lee, and Young-Koo Lee. Context awareness in large scale ubiquitous environments with a service oriented distributed middleware approach. In *Computer and Information Science, 2005. Fourth Annual ACIS International Conference on*, pages 513 – 518, 2005.

[3] J. Nakazawa and H. Tokuda. A middleware framework for sharing sensor nodes among smart spaces. In *Networked Sensing Systems, 2007. INSS '07. Fourth International Conference on*, pages 171 –178, june 2007.

[4] S. Chetan, J. Al-Muhtadi, R. Campbell, and M.D. Mickunas. Mobile gaia: a middleware for ad-hoc pervasive computing. In *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE*, pages 223 – 228, jan. 2005.

[5] J. King, R. Bose, Hen-I Yang, S. Pickles, and A. Helal. Atlas: A service-oriented sensor platform: Hardware and middleware to enable programmable pervasive spaces. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 630 –638, nov. 2006.

[6] D. Kulkarni and A. Tripathi. Application-level recovery mechanisms for context-aware pervasive computing. In *Reliable Distributed Systems, 2008. SRDS '08. IEEE Symposium on*, pages 13 –22, oct. 2008.

[7] A. Sutagundar, S.S. Manvi, Selvin, and M.N. Birje. Agent-based location aware services in wireless mobile networks. In *Telecommunications, 2007. AICT 2007. The Third Advanced International Conference on*, pages 20 –20, may 2007.

[8] A. Genco, S. Sorce, G. Reina, and G. Santoro. An agent-based service network for personal mobile devices. *Pervasive Computing, IEEE*, 5(2):54 – 61, april-june 2006.

[9] N. Miller and P. Steenkiste. Design and evaluation of a hybrid physical space service for pervasive computing applications. In *Mobile and Ubiquitous Systems: Networking Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on*, pages 1 –8, aug. 2007.

[10] Mohamed Ahmed, Hamid Harroud, Roger Impey, and Ahmed Karmouch. Agent-based multimedia presentation and adaptation service. *Multimedia Tools Appl.*, 34(3):299–315, 2007.

[11] Sadek Jbara, Tsvi Kuflik, Pnina Soffer, and Oliviero Stock. Context aware communication services in "active museums". In *Software-Science, Technology Engineering, 2007. SwSTE 2007. IEEE International Conference on*, pages 127 –135, oct. 2007.

[12] Bilal R. Qazi, Ángela A. de Grado Vivero, and Jaafar M. H. Elmirghani. Tina: The intelligent airport: a proposed architecture. In *IWCMC '09: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing*, pages 805–809, New York, NY, USA, 2009. ACM.

[13] E. Bruns, B. Brombach, T. Zeidler, and O. Bimber. Enabling mobile phones to support large-scale museum guidance. *Multimedia, IEEE*, 14(2):16 –25, april-june 2007.

[14] T. Miyashita, P. Meier, T. Tachikawa, S. Orlic, T. Eble, V. Scholz, A. Gapel, O. Gerl, S. Arnaudov, and S. Lieberknecht. An augmented reality museum guide. In *Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on*, pages 103 –106, sept. 2008.

[15] T. Kawashima, Jianhua Ma, Runhe Huang, and B.O. Apduhan. Gupss: A gateway-based ubiquitous platform for smart space. In *Computational Science and Engineering, 2009. CSE '09. International Conference on*, volume 2, pages 213 –220, aug. 2009.

[16] Open Mobile Alliance. Mobile location protocol. Candidate Version 3.1, 2004.

[17] Open Geospatial Consortium. OpenGIS Location Services (OpenLS): Core Services. OGC 05-016, Version 1.1, May 2005.

[18] Julie Pichon. Case Study: Implementing Location Services Based on Open Standards, 2008.

[19] Roy Fielding. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Informational), 1999.

[20] Henning Schulzrinne, Columbia University, A. Rao, Netscape, R. Lanphier, RealNetworks. Real Time Streaming Protocol. RFC 2326 (Informational), 1998.

[21] Ken Arnold, James Gosling, and David Holmes. *The Java Programming Language*. Addison-Wesley Professional, 2005.

[22] James Edward Keogh. *J2ME: the complete reference*. McGraw-Hill/OsborneK, 2003.

[23] Richard Matthew Stallman, Roland McGrath, and Paul Smith. *GNU Make*. Free Software Fundation, April 2006.