Towards the tailoring of a ubiquitous interactive model applied to the natural and cultural heritage of the Montsec area

Montserrat Sendín[1], Jesús Lorés[1], Jordi Solà[1]

[1] Computer Science and Industrial Engineering Department
University of Lleida, 69, Jaume II St., Lleida, SPAIN
Tel: +34 973 70 2 737 Fax: +34 973 702 702
e-mail: {msendin, jesus}@eup.udl.es
jordi@griho.net
URL: http://www.udl.es/dep/diei

Abstract. Starting from the multimedia information about the natural and cultural heritage from the Montsec area - that is already available on the interactive CD-ROM "La Memoria del Montsec" -, we propose an interactive model that offers such information with a double functionality. Firstly the diffusion via web, through as many internet devices as soon as they appear. The result is a multi-device dynamic web application which we have called "The Montsec Web". Secondly, in order to be used in situ, to be a location-aware system. Thus, another level of interactivity is added, obtaining a mobile and context-aware multimedia environment which we call "The Interactive Montsec". Both functionalities should be integrated on a common underlying technological base.

Additionally, the system will have to anticipate to the visitors' requirements, adapting the selected information to his changing profile and history.

Keywords. ubiquitous computing, location-aware systems, augmented world model, adaptive hypermedia, user modelling.

1 Introduction

The mountains range of the Montsec are situated in the western sector of the Catalan low Pyrenees (in the province of Lleida). It is an area of outstanding geological, historical, paleontological and scenic wealth, boasting a valuable natural and cultural heritage, which is why it is so highly regarded as a truly natural laboratory. The application of the model presented here in the Montsec area will help to conserve and make people more aware of the park's natural resources and cultural heritage all over the world.

The nowadays proliferation of Internet devices means that two-layer separation techniques – code and data – to create web pages will not be able to solve the problems that internet applications programmers are soon going to have. This is one of the reasons why it is being necessary to search for other technologies that allow to add another layer more in the applications development. It is the presentation layer, that is to say, the visual appearance that the final hypermedia document has to take. The result is a three-layer functioning structure: data, code and presentation. This is the structure we have adopted to our prototype.

With the technological base we propose here we aim to spread the wide information about the Montsec zone to two well-differentiated environments. The first is the own Montsec zone, focusing on the generic visitor's requirements arising throughout his itinerary. We are referring to “Interactive Montsec”, presented before. The other environment is the internet one, satisfying in this case to a generic net user who can use whatever existing internet device at present – referred before as “The Montsec Web” -. This last has to be possible without having to redesign the application. In both cases, the generic term used here makes reference to a user corresponding to a certain typology, which is gathered in the corresponding user modelling.

Our system solves the automatic generation of hypermedia documents in which the visual aspects related to documents presentation are based on previously designed templates – presentation models -. This avoids having to restrict documents presentation to an only particular knowledge representation. These templates are designed in order to adapt a same information to different devices. But this objective does not wholy cover all the expectations defined for the system.
The definition and processing and of the visual aspect separately will also solve the presentation of dynamic contents according to several changing contextual aspects. In the case of The Montsec Web these aspects are the user's profile and the kind of device used by him. In the case of The Interactive Montsec, the history of the visits and activities made by the user and, of course, the geographical location get also into consideration.
At the moment a prototype of The Montsec Web is already available as a multi-device dynamic application, in which the adaptation to different kinds of users and contextual situations should be improved. This means increasing the flexibility in several aspects. On the one hand, to set correctly a user modelling taking into consideration a bigger number of parameters. On the other hand, to offer the possibility of recording the evolution of each user along the time. Moreover, the user modelling constitutes a very important and determinant aspect to be considerated when we are working the visual aspect of documents.

In relation to the second pointed functionality, the location-aware systems, under the name of location based systems, have recently received great attention as application field for mobile systems, and they have a very promising future.
In order to offer location-awareness – for instance, to show the position where the user is on a map or to make spatial queries, such as the nearest points of interest in relation to the location we are -, it is required a detailed model from the real world as a base. This model has to be developed and integrated in the architecture [1]. To make this non trivial task easier our proposal consists of delegating this responsibility to a spatial data managing module. This module will be in charge of building that model and of making possible its access. This is the objective of the Nexus project: providing a detailed model of the real world suitable to the functionalities and features of each location-aware application in particular. This model is called Augmented

World model (AW-model), and it combines objects from the real world with virtual objects. The Nexus platform is being developed at present at the University of Stuttgart [2].

These spatial questions will consist of identifying and locating the nearest points of interest, whether historical, geological, geographical, biological or even tourist nature – these are the thematic sections in which the information is structured -. These points of interest will be showed selecting or priorising those that better adjust to a particular user's profile.

Under this expectation we can imagine the Montsec zone as a virtual museum where we can whenever obtain information from whatever is surrounding us in a implicit and tailored way. In an implicit way because the system solves automatically the location-awareness and, based on this it selects the relevant information anticipating so the visitor's requirements [3]. In a tailored way because the system takes into consideration contextual information related as much to the user's profile as to the visits and activities made by the user history, both of them brought up to date.

In relation to this second functionality we want to point out that a prototype based on GPS (Global Position System) has also been developed. According to the coordinates corresponding to the geographical point where we are, which are provided by a GPS device, the application shows the required information without having specifically to ask for it [4]. This prototype has been recorded on a videotape in November 2000 by the Catalan TV channel "Canal 33" [5].

The rest of the paper is made up of five sections more. On section 2 we present the technological structure that we have already developed up to date. In section 3 we expose the spatial and contextual data management that we propose to complete our Montsec web prototype. This way we intend our system to be anticipative and adaptive [6]. On section 4 we outline a sketch for the architecture which should integrate our starting system together the rest of the exposed functionalities. Next, on section 5, we present the tests we have carried out with the prototype. And finally, the conclusions.

2 Technological Structure of The Montsec Web Prototype

2.1 Layered Structure

As has been pointed previously, the application is structured in three layers which interact each other as we can see on the figure 1.

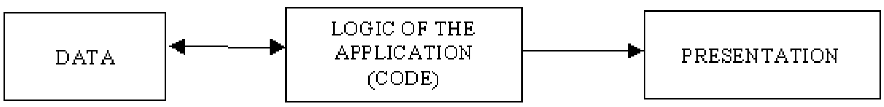

Fig. 1. Layered Structure

Let us see the aim of each layer.

Data layer. It is made up of a relational data base that at first gathers all the tuples of information selected for the prototype. By means of a JDBC driver we make possible to connect Java to the data base. In short, this layer provides required data to the code layer.

Code layer. This layer starts the processing of the request by sending a SQL query to the data base. When it receives asked data in jsp format, it translate them to XML format, sending the resulting XML file to the presentation layer which will have been validated and it will be well-formed.

Presentation layer. The aim of this layer is to translate data received from the code layer to different presentation languages, suitable to the current device (HTML, DHTML, WML, and FO among others). The code translation is carried out using some previously designed templates in XSL[1] code (eXtensible Style Language). It exists one template for each internet device we have planned. These templates combine the device specific code with XSL language labels. This labels are replaced by data extracted from the XML file at run time, in order to generate the corresponding presentation.

2.2 Underlying Technology

Now we want to present the operations to carry out on the code and presentation layers in more detail, showing how these are carried out and how they interlace each other. In addition, we want to detail the chosen technology for each part.

Data conversion to XML format (Code layer). The SQL query to the data base provides data in jsp format. The first step to do consists of converting this data to XML code, as we have already said.
Such a query is made by means of a jsp file. This file includes XML and JSP labels, as well as predefined labels of Cocoon[2] too. These labels basically show which template has to be used in the presentation layer to adapt the final web page to the kind of navigator that made the request.
The treatment of jsp code is carried out by the JSP servlet. It replaces the pertinent lines by data received from the data base. The result is the expected XML file. The JSP servlet referred resides in the Orion server we have used.

Validation and parser of the XML code (Code layer). The XML code obtained in the previous step has to be validated by means of a text file called DTD. The result is

---

[1] It is a language of styles to give format to XML data documents.
[2] It is a frame to publish web pages based 100**%** on Java and in the last W3C (World Wide Web Consortium) specifications [7] as DOM, SML and XSL.

a valid and well-formed XML file. Namely we have used the XERCES version 2.0 parser, from the Apache company.

Selection of the XSL page (Presentation layer). The selection of the XSL template to apply is worked out by the Cocoon (version 1.0.) module, which selects the one appropriate to the user's device. This module has been configured as a filter servlet. In that way, we achieve that the Cocoon servlet acts later than the JSP servlet. We want to underline that a filter is basically the main component which makes possible to combine the different technologies we have used. In particular, this was one of the most important requirements that the web server had to fulfil [4].

Data conversion to the final format (Presentation layer). The last step to realise after having the XML code validated and the XSL presentation template selected is to translate definitely data to a code directly treatable by the client navigator. Finally this resulting page will be returned to the client.
The XSLT[3] (conversions language) converter used has been XALAN version 2.0.
On figure 2 we can see how everything gets together.

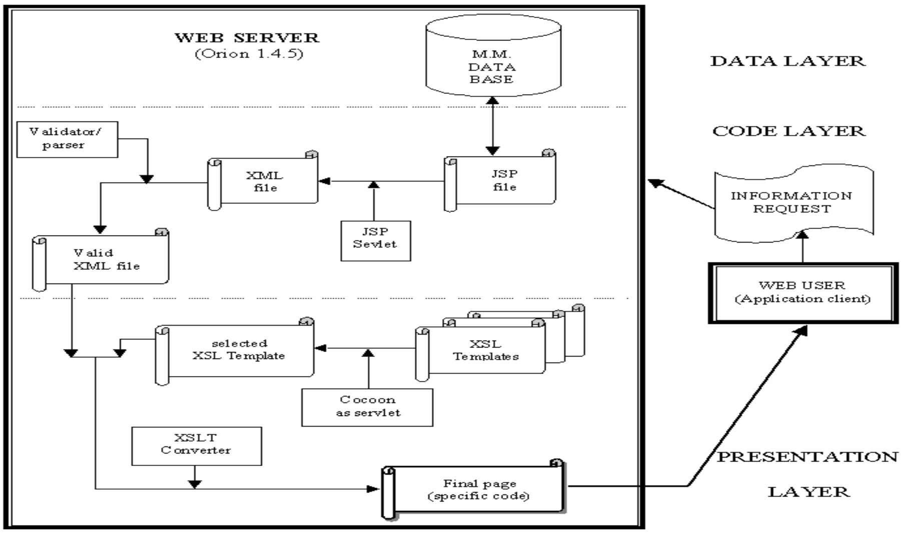

Fig. 2. Prototype technological structure.

---

[3] It is a much more practical extension of XSL that allow us conversions of XML data to different formats.

3 Spatial Data Management and Tailoring

3.1 Augmented World Model Accessibility and Characteristics

The detailed model of the real world constitutes the central information model that provides an integrated and homogeneous view on the data available at each moment, in order to offer location-awareness [8]. Developing a detailed model of the real world turns out a costly task, specially if this one has to be updated, as it happens in our case.
Up to now, all the location-aware applications use their own particular world model. The goal of the Nexus project is to provide a detailed model of the real world suitable to location-aware applications, according to their concrete functionality – both for indoor and outdoor usage [9] -.
The result consists of representatives for real world objects and of virtual objects that provide, among other things, links to external information spaces like the web. This model is called the Augmented World Model (AW-model).
This model is kept up to date by integrating the update of sensor systems, for example to obtain the current position of mobile objects. In our case this updating is limited to know the position more o less approached where the visitor is at any moment. This one will be provided by the available GPS device.
Therefore, our application will formulate a query related to the visitor's zone. The Nexus platform will process this query accessing to the AW-model, and returning the requested information again to the application.

Location-aware applications may query the current state of the model by using the Augmented World Querying Language (AWQL) and receive as answer information about the model described by the Augmented World Modelling Language (AWML).
As much AWQL as AWML are languages based on XML. As we are working in XML format too, the compatibility is guaranteed. All that make us to think that the integration of the Nexus platform to our architecture will be feasible.

The Nexus platform basic interactions are graphically represented on figure 3. This interface offers to typical location-aware applications the required functionality hiding the details of the underlying data management. We should envision it as a middleware that brings together different providers and consumers of location-based information [2].

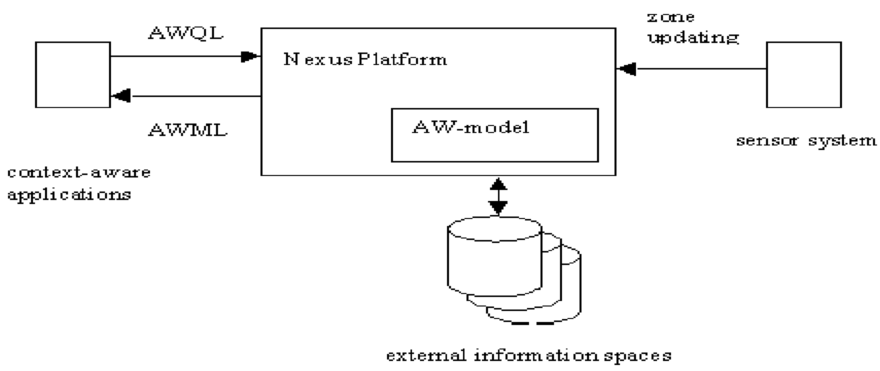

Fig. 3. Basic interactions of the Nexus platform

As the Nexus platform architecture – at the moment being in construction by the research group devoted to the Nexus project (Sttutgart University) -, does not constitute the central subject of our paper, it will not be showed in more detail.

3.2 Knowledge Representation

The essential components of the AW-model are the static, geographical objects data (spaces model: castles, towers, monasteries, rural areas, churches, etc), the location information of mobile objects (model of mobile objects: the zone visitors), and the virtual objects which provide additional information or services to Nexus users. The late have no counterpart within the real world, but they can be visualised by an appropriate application.
In order to define the AW-model, the Nexus platform uses an object-oriented approach, which allows the AW-model to be easily adapted to each concrete case. There are two categories of classes: members of the Standard Class Schema and the Extended Class Schema. The Standard Class Schema contains classes that we consider fundamental and that have a well known structure, so they can be used by every application. The Extended Class Schema will contain additional classes for special purposes, suitable to a concrete location-aware application.
It is aimed to identify the set of classes that better fits a certain geographical application area. This idea corresponds to the ontology concept.

The domain classes will be represented in AWML. So, the answer to an AW-model query will consist of a list of matching objects, in a particular zone. For example, a list of the nearest restaurants with their respective attributes – address and menu, for instance – and also a list of the castles feasible to visit in the zone, according to attributes as historical period and visits timetable. This would be suitable if our user has a certain preference for History. Depending on the interest degree – being a lover or an amateur – different information levels could get in action.

The restaurant object and the castle object are two examples of suitable entities of the Montsec area. So they have to be part of the Extended Class Schema. Another option would be objects that specialise some object from the Standard Class Schema with the suitable additional attributes, if it is possible.

The information level required by the user, or if, for example, he knows already a term related to the information such as the features of the romanic style of a church - these are aspects that we refer as contextual information -, will also have to be considered and modelled during the formulation of the query. We will so insert simple expressions on the query specification, for instance in Java. A possibility is to use an HTML extension based on JSP.
To define and formulate the query of this kind of adaptive elements in the AW-model we are going to introduce conditions on whatever class from the classes hierarchy. These conditions will have to match with the Java expressions included in the query. In that way it will be possible to construct dynamic structures which will take their definitive form at run time, according to the user model [10].

The contextual information will be used at run time to adapt to the user both the contents and the links selected – previous to the query time – and his presentation – at the generation of the hypermedia document time -.

3.3 Presentation Model

In The Montsec Web prototype developed we have only considered the device ubiquity. That is why designing a template suitable to each internet device was enough.
To also integrate the contextual information on the presentation – adaptation to the user's features and current situation – we have to diverse and extend the possibilities to generate a final presentation. We have to adapt the presentation to several circumstances, as much as possible [11].

In order to carry it out we are going to adapt two measures. On the one hand, we will enlarge the number of previously designed templates, taking into consideration not only the type of device, but also, for example, the preferred thematic section, according to the user's preferences. We consider this factor determinant in order to design the presentation.
On the second hand, we want also to update the templates contents. To generate personalised information we are going to introduce adaptive presentation elements, which are called presentation rules. They control aspects as the links generation, the spatial layout of lists, etc, and mainly take into consideration all kind of contextual information.

The combination of the templates language with the presentation rules will let us specify a wide set of non trivial presentations by means of a very simple syntax [10].

4 Integrating Architecture Sketch

We intend to integrate in our three-layer structured prototype the spatial data management proposed here to obtain a detailed model of the real world according to the purpose of our application, without having to carry out its management. This solves one of the functionalities that the starting prototype did not cover: being a location-aware application.

On the other hand, our final architecture has also to integrate all the aspects related to the contents adaptation according to the user's profile and history – as we call contextual information -. We will add two relational tables: the histories table and the profiles table, both of them indexed by the user's identification field.
As the contextual aspects will work out the query to realise, they will be solved a priori in order to outline the AWQL file used to make the query to the AW-model. Here is where the Nexus platform takes part. Additionally, the architecture has to be prepared to update as much the user's history as the user's profile according to the queries, visits or activities realised by the user, the concepts the user is assimilating and the preferences he is refining.
The identification module will solve all this management related to the contextual information. Later, on the presentation layer, it will operate again to select the template according to the thematic section preferred by the user. As we have pointed before, the offer of templates is now bigger, and in addition, their contents are adaptive.
Our aim is to obtain the adaptation degree that is proposed here, and to cover all the exposed expectatives.

In relation to the client of the application (the user), now he has a GPS device, and so he provides the information of the geographical coordinates, besides of his identification.

All that appears reflected on figure 4.

5 Tests Realised to The Montsec Web Prototype

We want to point out here that due to the developed application is only a prototype, the data base built is composed of a reduced number of tuples selected from each thematic section. Each tuple has been prepared to be consulted under three planned levels of information: basic, intermediate and expert. Besides, all that is available in two languages: Castilian and Catalan.

5.1 General Performance Tests

We have carried out part of the tests that usually are carried out for web applications. Our aim has been to validate on the one hand the robustness of all the code modules that take part and also the server. On the other hand, we have also validated the runtime efficiency. Finally, we have checked the simplicity of maintenance. Let us see them separately.

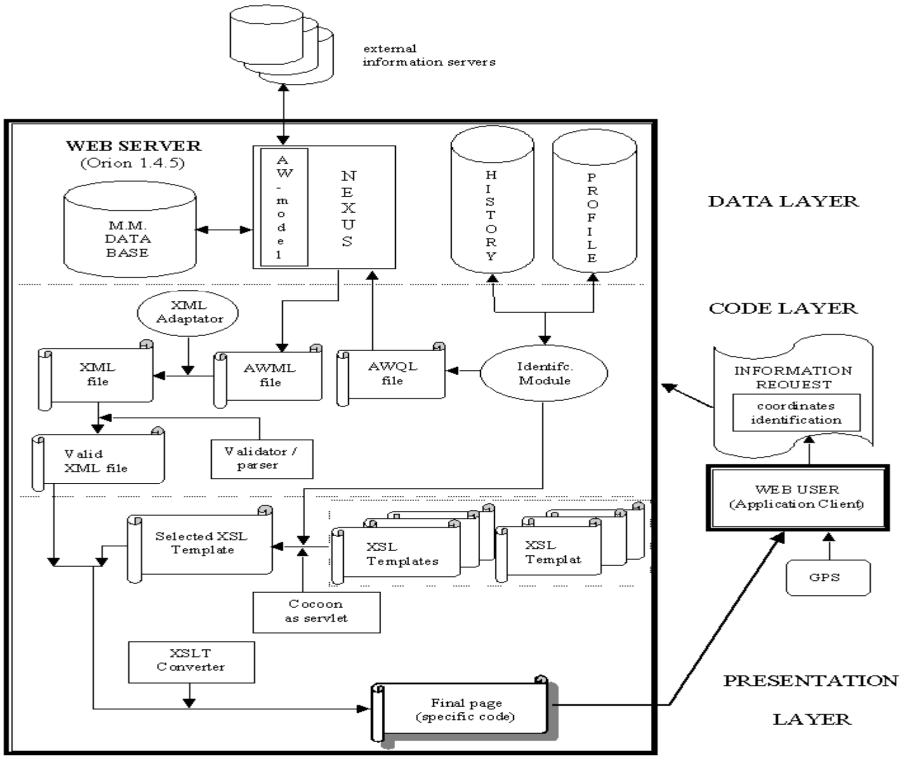

Fig. 4. Integrating architecture.

Server reliability. We have analysed the server efficiency in serving web pages and also the performance and mutual interaction among the servlets and the rest of the modules. We have made different tests with HTML, WML, JSP, CHTML and DHTML pages, and we have not detected any problem serving the different kind of pages.

In relation to the used parsers, validators and XML converters – XERCES and XALAN -, they have proved a high compatibility degree with the other modules of the web application, increasing with this the server reliability.

Download speed. We have evaluated this parameter in the different available devices. In general, the download speed has no been the expected one. Nevertheless, we have to take into account that the pages have to cross a sequence of stages (parse, validation and conversion from XML code) which implies a considerable and insuperable delay. With reference to that, we are expecting the version 2.0 of Cocoon in a short time. This will warrant an upper download speed.

Maintenance tests. The purpose of this kind of tests has been to prove the independence of the data layer from the code and presentation ones. In effect, thanks to the use of XML, no change has affected the application performance in any of the different versions of the presentations. So the tests have been fully satisfactory, as it was expected. Therefore we can conclude that our application behaves as a dynamic web.

5.2 Tests on Diverse Devices

It is obvious that the visual appearance and the format of presentation varies for each device, due to the graphic design and the navigation system have to be adapted to concrete possibilities of different devices performing the application.
Here we detail the different devices we have used to prove the application.

WAP devices. This test has been realised on a WAP mobile emulator. As the possibilities of contents visualisation allowed by these kind of devices are very limited, we have offered the information only on the basic information level, as it is the most reduced.

IMODE mobiles. IMODE is another internet access protocol for mobile devices, developed by the Japanese company DoCOMo. The required language is CHTML.
We have also used an IMODE mobiles emulator: the Internet Mobile Explorer, from the Microsoft company. This emulator also simulates aspects of download speed and screen visualisation capacity, which have also turned out useful.
The tuples visualisation in this tests has been right. Nevertheless, it has been necessary to use scroll bars to see a whole tuple, due to the reduced size of the screen.
The test realised with IMODE mobiles let us get a quite exact approach about the capacity of third generation mobiles.

Desktop Personal Computers. In this case, we have made two tests. The first one on one of the most used navigators: the Microsoft Internet Explorer version 5.5. The required language to prepare this kind of templates is DHTML.
The second test has consisted of converting different information tuples on PDF binary format, to be visualised with Adobe Acrobat Reader. In this case the required language is FO.

6 Conclusions

In this paper we present the ubiquitous interactive model that we are developing. This model will solve diverse interactivity levels and it will also fulfil several challenger features; among them to be adaptive and anticipative.
Applying this model to the Montsec environment will help to preserve and to promote all over the world the real state and natural and cultural resources of the park. On the one hand, it will offer adapted information to the zone visitors, obtaining this way a true “interactive space”, just as Weiser defined the ubiquitous computing in [12]. On

the other hand, it will also offer Montsec information to any net user from anywhere in the planet, without the device used offering any barrier.

The technological base obtained and presented in this paper turns out very flexible, adaptable and present, since it uses future technologies. The three-layer structure makes easier the reusability, modularity and consistency of the presentation, reducing in consequence the developing cost.

At present it exists a multi-device software prototype of The Montsec Web, which works with different templates suitable to each device, just as it was proposed in [13]. Nevertheless, all the aspects related to the user, which are going to be gathered on the user model, will be also incorporated to take part on the presentation dynamic generation.
The construction and use of an abstract presentation model taking into consideration all these aspects, will allow us to configure the adaptive presentation of contents independently of their developing [14].

Our future plans consist of constructing an architecture that incorporates, articulates and coordinates the device ubiquity as we have already solved, the location-awareness as we have presented here and finally the user modelling. Our final aim is to generate automatically hypermedia documents highly adaptive and tailored, using the technological base and the three-layer structure showed in this paper.

References

1. Cheverst, K., Davies, N., Mitchell, K., Friday, A.: Experiences of Develoing and Deploying a Context-Aware Tourist Guide: The GUIDE Project. Proceedings of the Sixth International Conference on Mobile Computing and Networking (MobiCom 2000), ACM Press (2000) 20-31

2. Grossmann, M., Leonhardi, A., Mitschang, B., Rothermel, k.: A World Model for Location-Aware Systems. Published on behalf of CEPIS by Novática and Informatik/Informatique http://www.upgrade-cepis.org., Vol.II, Issue no. 5. Ubiquitous Computing (2001) 32-35

3. Brogni, A.: An Interactive System for the Presentation of a Virtual Egyptian Flute in a Real Museum. Virtual reality in archaeology. Archaeopress (2000).

4. Solà, J.: El Montsec Interactiu. Career Final Proyect, Polithechnics Universitarian School, University of Lleida. (2001)

5. Sendín, M., Lorés, J., Balaguer, A., Aguiló, C.: Un Modelo Interactivo Ubícuo aplicado al Patrimonio Natural y Cultural del área del Montsec. Proceedings of the Second International Congress on Human Computer Interaction – Interaction'2001. Abascal, García and Gil (Eds.), Salamanca (Spain) (2001) 51

6. Pressman, R. S.: Ingeniería del Software. Un enfoque práctico. McGraw Hill, 4th edition (1997)

7. World Wide Web Consortium. W3C Note 08 May 2000, http://www.w3.org (2000)

8. Nicklas, D., Groámann, M., Schwarz, T., Volz, S., Mitschang, B.: A Model-Based, Open Architecture for Mobile, Spatially Aware Applications. Proceedings of the 7th International Symposium on Spatial and Temporal Databases (SSTD'2001) (2001)

9. Hohl, F., Kubach, U., Leonhardi, A., Rothermel, K., Schwehm, M.: Next Century Challenges: Nexus – An Open Global Infrastructure for Spatial-Aware Applications. Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, Washington, ACM Press (1999) 249-255

10. Castells, P., Macías, J.A.: Un Sistema de Presentación Dinámica Hipermedia para Representaciones Personalizadas del Conocimiento. Proceedings of the Second International Congress on Human Computer Interaction – Interaction'2001. Abascal, García and Gil (Eds.), Salamanca (Spain) (2001) 45

11. Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia.. Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers. Brusilovsky, Kobsa y Vassileva (Eds.) (1998) 1-43

12. Weiser, M.: Some Computer Science issues in Ubiquitous Computing. Communications of the ACM, Vol. 36, No. 7 (1993)

13. Sendín, M., Lorés, J., Aguiló, C., Balaguer, A.: A Ubiquitous Interactive Computing Model applied to the Natural and Cultural Heritage of the Montsec Area. Published on behalf of CEPIS by Novática and Informatik/Informatique http://www.upgrade-cepis.org., Vol.II, Issue no. 5. Ubiquitous Computing (2001) 23-26

14. Macías, J.A., Castells P.: A Generic Presentation Modeling System for Adaptive Web-Based Instructional Applications. ACM Conference on Human Factors in Computing Systems (CHI'2001). Extended Abstracts. Seattle, Washington (2001)