DEPARTAMENTO DE INFORMÁTICA

FRAMEWORK FOR UBIQUITOUS AND VOICE ENABLED WEB APPLICATIONS DEVELOPMENT: A TRANSPORT INFORMATION SYSTEMS TEST - BED

EDUARDO CARRILLO ZAMBRANO

UNIVERSITAT DE VALENCIA
Servei de Publicacions
2004

Aquesta Tesi Doctoral va ser presentada a Valencia el día 6 de Setembre de 2004 davant un tribunal format per:

- D. Gregorio Martín Quetglas
- D. Santiago Felici Castell
- D. Mark Baker
- D. Manuel J. Fernández Iglesias
- D. Ismail Khalil Ibrahim

Va ser dirigida per:
D. Juan José Martínez Durá

Depòsit legal:
I.S.B.N.:84-370-6063-X
Edita: Universitat de València
Servei de Publicacions
C/ Artes Gráficas, 13 bajo
46010 València
Spain
Telèfon: 963864115

FRAMEWORK FOR UBIQUITOUS AND VOICE ENABLED WEB APPLICATIONS DEVELOPMENT:

A Transport Information Systems Test-Bed

Doctoral Thesis Dissertation
European Doctorate Modality

Eduardo Carrillo Zambrano

Supervisor: Dr. Juan José Martínez-Durá

Departamento de Informática
Universidad de Valencia
Valencia
Spain
2004

DEDICATION

This thesis dissertation is dedicated with great love to my wife and my daughter from whom I have been separated for more than 3 years; during this time I have received infinite patience and understanding.

Also to my parents and sisters, who never doubted me for a moment and who also have given me the strength to reach this objective.

AKNOWLEDGEMENTS

From the moment I started creating the list of names of whom I wished to thank, I knew that someone would be missing from the list, from among so many who have collaborated in the realization of my work; but I think it would be insincere and rather superficial if I did not at least try.

The culmination of this thesis has been possible thanks to the participation of a great number of professors, colleagues, students, relatives and friends with whom in one way or another I have had a relation before and during my work, from now on referred to simply as "people".

Giving it a second try, I want to express my most sincere gratitude to all the people with whom I have interacted at the Universities where I have spent the last years trying to train as an investigator and teacher, that is to say the Universidad Autónoma de Bucaramanga (UNAB), Colombia, and the Universitat de Valencia, Spain, my second country.

I would also like to thank the people with whom I have been in contact at the Universities where I carried out my investigation in the final part of my thesis, such as the Group of Distributed Systems at the University of Portsmouth, England, as well as the Department of Telecooperation from the Johannes Kepler University of Linz, Austria.

I consider that in this first global intent of expressing my gratitude, nobody will be left out of the list even though their names is not explicitly mentioned.

Finally I want to give special thanks to:

- Doctor Gabriel Burgos Mantilla, rector of the UNAB, who gave me the possibility of starting my search of new horizons of investigation and who gave me the confidence of having employment at the end of my training.

- Cesar Dario Guerrero Santander, Computational Science professor and Dean of the Systems Engineering Faculty of UNAB, who trusted in me when proposing me as a candidate to commence my studies in Valencia with the sponsorship of UNAB.

- Doctor Gregorio Martín Quetglás, Director of the Robotics Institute of the University of Valencia, who has given me his constant support for the continuity of my education as an associated investigator of the institute.

- Doctor Juan José Martínez-Durá, my thesis supervisor, who gave me not only the initial entry into the Institute under his supervision, but also allowed me to have continuity during the time I have been part of his group, offering me outstanding conditions for the realization of my work, and allowing me to participate in a selective manner and under mutual agreement in projects that were of interest to me.

- Doctor Santiago Felici Castell, who has known how to guide me in the complex decisions I have had to make, not only at levels of doctorate education but also on a personal level.

- Doctor Mark Butler, Hewlett Packard investigator and creator of the DELI library; he has given me important ideas in the process of defining the line of investigation that I could follow in my thesis.

- Ph.D. (c) José Javier Samper Zapater, who has given me his friendship and has trusted in me to become a member of his work group, and who has put up with my indecisions and my constant changes of direction.

- Doctor Juan De Mata Domingo Esteve, one of the most prestigious investigators at the University of Valencia, who guided me with valuable suggestions and who assisted me from the beginning in my search for scholarships that would allow the continuity of my work, with his strategy of "ask for as many scholarships as possible, even though you might have to turn one down".

- Engineer Pablo Catalá Varas, the first great friend I made in Valencia, who besides having offered his friendship to me, has known how to bring into reality the graphic

ideas I have proposed on paper, with results that are in many of my international publications as well as in this work.

- Engineer Arturo Cervera García, my colleague and friend, the best quality software developer I have met in the almost 14 years I have been in this area, who has given me all his support at all times and from whom I have also learned a lot.

- To those who at different times have collaborated in the grammatical revision of the English translations of my several publications, Karin Buttner, Ana Cabrejas, Leo van den Berg and Cristina Rodas

- To my wife Yamile Gómez Aguilar, to whom besides dedicating this work I must give thanks, because she has known how to put up with all those moments of crisis that many of my friends can't even imagine I've had, and who has given me her love and encouragement to continue until reaching the goal we have proposed to ourselves.

- To my daughter Leidy Juliana Carrillo Gómez, who regardless of her young age and in her own way has known how to give me words of encouragement, and has motivated me to study "the way she does" with responsibility and enthusiasm. Hopefully one day her dream of obtaining a doctorate scholarship that includes her mommy and daddy will come true.

Of course I give thanks to God, that God that exists with different names in so many cultures, but in the end is the same one who is present in every living being around us, and has given me the wisdom and patience to be able to culminate this work.

I have learned much from all the people that I have mentioned in a global or specific way; they have helped me to grow, not only at a professional level, but more importantly as a human being. To all of them "thank you".

ABSTRACT

The purpose of the Ubiquitous or Pervasive Computing - an emergent paradigm of personalized computation- is to obtain device interoperability under different conditions. The devices were designed for different purposes by different companies or from different "technological" generations.

The ever increasing market of "web enabled" devices has brought up diverse challenges related to the difficulty of visualizing content in a unified form to diverse clients, while at the same time taking into account the great differences in the capacities of these devices. It is not feasible to develop a separate application for each of these devices, simply because the number of different devices is too high and still growing.

In the analysis of existing proposals dealing with the modelling of ubiquitous web applications, the link that exists between the logical and conceptual modelling and the physical modelling of the applications is not clear enough, and the way in which the context aspects related to web access from these devices cannot be specified. On the other hand, the available commercial products are supplier-specific. Every future platform change would a costly and painstaking process

In this thesis we present a proposal of a framework for the development of web applications that can be accessed from different types of devices, such as PCs, PDAs, mobile phones based on diverse technologies (like WAP and I-Mode) and conventional telephones that access the web through voice gateways and voice portals.

The proposed framework serves as a guide for the development of this type of applications and it can be deployed to different server configurations and software development technologies.

In order to obtain this objective, a description of diverse theoretical elements related to dynamic generation of information that can be acceded by devices has been made,

as well as a description of involved technologies whose hardware, software and connectivity characteristics vary remarkably.

The theoretical study was carried out in parallel with tests based on the different technologies used. A multilingual-ubiquitous traffic information portal was used to test the theory in an operational environment.

INDEX

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Background

The number of internet-enabled devices is growing rapidly. At present there are mobile phones based on different technologies such as WAP (Wireless Application Protocol), I-Mode, Personal Digital Assistants, Internet Kiosks, conventional phones, interactive television devices and other types of appliances.

In a recent survey [Vol04] some 600 different devices were identified and this number continues to grow. In addition some 200 different properties were identified for each device, including the types of mark-up languages, security capacities, input and memory capacities, supported media types, mail capacities, network connectivity, etc.

For hereinafter, the term "web application" will be used for applications that can be executed in a Web-based environment. A ubiquitous [1]or pervasive web application will be a Web application that can be executed anytime/anywhere/any media.[SchO1].

Pervasive or ubiquitous computing is based on the study of the convergence between the Internet and advanced electronic technology, in particular wireless access available to a number of devices.

Ubiquitous web applications have diverse differentiating elements in relation to the traditional applications. Aspects, such as the mobility that is associated with this type of applications, and the limitations and diversity of existing characteristics in the devices, constitute context elements that differentiate them.

Mobile devices allowing Internet access includes a micro browser that permits web navigation. Web access can be obtained by using diverse mechanisms (intermediate gateways, wireless networks available to mobile telephones, through synchronization with PCs).

[1] The term ubiquitous computing was introduced in 1991 by the investigator of Xerox Mark Weiser.

At present there are different proposals for the logical and conceptual modelling of Web applications that can be expanded to the modelling of ubiquitous web applications.

However, there is a lack of clarity in the approach to physical modelling. It must be based on the detection of a minimum set of internal elements, and it must be present in this type of development bearing in mind the different capacities of these applications.

The research performed in this thesis was done in the Intelligent Transport Systems (ITS) group, which is part of the Robotics Institute of the University of Valencia. This group particularly specialises in system development based on emerging technologies in ITS.

This thesis investigates the specific activities of Web-based Traffic Information Systems for the European macro-projects ARTS (Advanced Road Telematics in the South-West) and SERTI (South European Road Telematics Implementation), co financed by the European Union.

Problem Description

The basic problem is that the Internet today is designed to be accessed from a PC. Most content assumes a high-resolution screen with sophisticated graphical capabilities.

Anyone who has tried to access the web from a wireless connected PDA knows that either it does not work well because the PDA browser cannot handle a complex page, or that after a painfully long loading time, the page cannot be properly rendered because of the PDA's small screen.

Mobile phones exhibit another problem, common to many PDAs too, the lack of a keyboard. This makes difficult entering information. The most frequently need required when browsing the Internet is typing of URLs. Doing this on a mobile device or with handwriting recognition systems is challenging, to say the least.

To develop an application for each of these web-enabled devices separately is *not feasible due to the high developmental costs involved.* However, to develop a generic web application that can use the maximum capability of the different devices constitutes a complex task

Thus, we have to take into account that there are no reference guides for developing these web applications, that extends beyond models of logical components or style guides for the development of interfaces. Although important, they do not constitute a framework to approach the problem.

In this sense, this thesis focuses on physical modelling, and it can be included with other existing alternatives for logical and conceptual modelling or as a starting point for a proposal that includes logical and physical modelling of ubiquitous web applications.

Therefore, the research questions that have arisen are:

- What mechanisms or proposals can be developed to reach an efficient handling of information in the development of ubiquitous web applications?

- Is it possible to obtain a reference framework that helps to approach the problem of the device independent content generation?

- Is it possible to recommend a scalable and vendor-independent solution?

Objectives

Based on the aforementioned research questions, the objectives that were considered in this thesis are as follows:

- Main Objective

- To propose a framework, serving as a generic work guide, for the development of Web applications that can be accessed by different devices, which avoids the development of a separate web application for each device type and considers capacity variations.

- Specific Objectives

- To validate the proposal for a multi-language traffic incident information web application accessible from mobile phones, PDAs and conventional telephones.

- To propose grouping of device models, which share common generic characteristics.

- To propose reference architecture for the development of ubiquitous web applications on dynamic interaction, independent of specific server technologies, which allows the incorporation of grouping as previously mentioned.

- To propose an integrated development environment which facilitates the editing of device specific documents, and which can be used with different technologies (from script development to server configuration), including the possibility of generating scripts needed for the process of transformation to adapt to different formats

- Methodology

The developed research is of an exploratory character and it has been supported by a wide and permanent revision of the theoretical basis, of the involved technologies and the development of the state of the art of ubiquitous Web applications. Furthermore the research has been based on the evaluation and application of the technologies studied in the field of Intelligent Traffic Systems.

The theoretical framework and the state of the art were developed taking as basis books, scientific publications and documents of European projects available from the research team, training courses and Internet searching.

The theoretical study was carried out together with some tests based on the different technologies involved in the solution of the problem and the need to set up real traffic information portals that could be accessed from mobile devices.

Some of the results were checked by discussions held both in seminars with the research team and by participating in different conferences. Also some research visits were made, where the tests developed during the work were discussed and analysed obtaining some very valuable contributions and criticisms.

At the same time as the research was being developed it was proposed that some students should present their degree projects on the subject. These degree projects helped support some partial implementations related to the work and additionally information about the advance of the work was distributed to several training researchers.

Organization of the Document

To respond to the previous objectives, the dissertation has been organized in three major sections:

In the first section the theoretical frame, the technologies involved and the state-of-the-art, are described.
Chapter 2 describes the mechanisms that allow connecting to the Web with different device types, the communication protocols involved (WAP and I-Mode), the mechanisms of accessing the Web from PDAs and voice portals. Also included is a description of the evolution of mark-up languages associated with the navigation by different devices.

Chapter 3 describes the theoretical framework and state-of-the-art related to the development of ubiquitous web applications, taking into account the efforts of

communities such us Software Engineering as well as of the perspective of the Knowledge Representation. After that the W3C's approach towards device independence is presented, as well as open-source and commercial solutions based on these proposals.

Chapter 4 ends part one. This chapter starts with definitions related to Intelligent Transport Systems. After that a detailed description is included of the full range of possible functions in transport logistics that can be addressed by transport telematics technologies, as well as its benefits.

In the second sections the main proposals and contributions are described.

This section presents the contribution based on the framework for development of ubiquitous web applications

Chapter 5, describes the proposal for characterization of groups of devices that avoid the development of multiple web applications.

Chapter 6, describes a generic architecture that allows platform and technology independence. The implementation and testing stages of the framework are also described.

Part one ends with chapter 7. Chapter 7 presents the integrated development environment for editing applications. It is based on the proposals for device independence.

The third section discusses the results from tests developed during the research process and the validation of the framework

Chapter 8, provides a description of different prototypes and applications for Mobile Internet that were developed at the same time as the theoretical revision process, as well as the initial tests based on device independence solutions.

Chapter 9, describes the implementation of a ubiquitous web application based on traffic incident information all of which serves to validate the framework proposed in this thesis.

Finally the conclusions reached and the recommendations for possible future research are described. The final section provides the bibliographical references as well as publications derived from the research performed.

SECTION I: THEORETICAL FOUNDATIONS, ENABLING TECHNOLOGIES AND STATE OF THE ART

2. PERVASIVE OR UBIQUITOUS COMPUTING

2.1 SUMMARY

In this chapter, the main protocols that allow wireless Internet access to different types of devices are described, as well as markup languages used in the content editing processes; finally, the concept of context and its relation to the development of mobile applications is presented.

Within the protocols for mobile devices, the characteristics and internal structures for WAP and I-Mode protocols are presented, as well as the different possibilities that can be used to obtain PDA connectivity.

Afterwards, the way in which conventional telephones can obtain access to Web content by means of voice portals is detailed, and the components and interaction of this type of portal is described.

Additionally, an outline of the evolution of markup languages and a description of those that have been derived from XML and that are used in the content development for mobile telephones are presented.

Finally, different definitions of context that have been generated by diverse communities are presented, such as man-machine interfaces and hypermedia, from which definitions of this concept have derived and which have been used in the development of mobile Internet applications.

This context conceptualisation will serve as the base for the development of definitions that will be found in the following chapter, such as the Delivery Context, which in turn forms part of the nucleus of the device independence principles that serve as guide to the development of ubiquitous applications.

2.2. INTRODUCTION

The traditional access to Internet based on the use of a computer connected to the network has characterized by limitations related to the restrictions in mobility and the number of users who can have access to it. [Des01]

With the evolution of the technology, this access to the Internet has been allowed from diverse types of devices: mobile phones, personal digital assistant, conventional telephones with access to voice enabled web pages, etc; and one hopes that in a future this capacity is integrated in many more devices like household appliances. This has generated a new range of possibilities for the development of technological solutions based on web applications that can be useful to new users of the Internet.

The differences in the protocols that govern the communications in wireless Internet are well known. On one hand the model WAP (Wireless Application Protocol) that predominates in Europe and America, is quite different from the I-Mode model, developed by the Japanese company NTT-Docomo.

Additionally, the adoption of the technologies of second and third generation has allowed to overcome the main disadvantages that displayed the first generation of WAP applications, where the slowness of the service was due mainly to the wireless technology that governed the wireless communications, i.e. GSM (Global System for Mobile Communications) [Rab01]

In this sense the development of prepared applications for the Internet has found new challenges related mainly to the search of efficient mechanisms for the design and implementation of adaptable Web sites to these new requirements.

Different mobile devices support different characteristics from HTML (Hypertext Markup Language) or use other markup languages, for example, most of telephones WAP use WML, while the I-Mode telephones use CHTML, a simplified version of HTML.

On the other hand, the developers are facing several serious problems: the constant change in the markup languages, the lack of standardization of them, variations in protocols on which their applications must be executed (even the coexistence of many protocols in the same place), difficulty of making tests on a great number of existing devices (since the available emulators show differences with respect to real devices). All these problems must be solved under the perspective of avoiding the development of different content for each of the available terminals

Additionally the hardware characteristics of the devices can be remarkably different. The input devices can be based on keyboard, touch screens, voice, etc. The output can be made by screen, by means of sounds, in printed format, etc. Also, the differences in processor, memory and battery are great between the different models.

This is why one of the main challenges that at the present time is to generate content that can be adapted to different types of devices and the preferences of the users.

The Ubiquitous Computing-Pervasive Computing is an emergent paradigm of the personal computation that is characterized to switch from mechanisms of computation dedicated to the use of capacities of generalized computation, which they are in our daily life. Some characteristics of these new mechanisms are that they are small, portable and apt to be used within wireless networks. The general platforms and the devices of wireless nature require of a special architecture and they are the foundation of this area of study.

The key goal of ubiquitous computing is interoperability under "unchoreographed" conditions, i.e., devices which didn't necessarily designed to work together (such as ones built for different purposes, by different manufacturers, at a different time, etc.) should be able to discover each others' functionality and be able to take advantage of it. Being able to "understand" other devices, and reason about their services/functionality is necessary, since full-blown ubiquitous computing scenarios will involve dozens if not hundreds of devices, and a priori standardizing of the usage scenarios is an unmanageable task [W3C03d]

For some, Ubiquitous computing is about mobile data access and the mechanisms needed to support a community of nomadic users. For others, the emphasis is on "smart" or "active" spaces, context awareness, and the way people use devices to interact with the environment. And still others maintain a device-centric view, focusing on how best to deploy new functions on a device, exploiting its interface modalities for a specific task. [Cun01]

Ubiquitous computing encompasses all of these areas, but at its core, it is about three things. First, it concerns the way people view mobile computing devices, and use them within their environments to perform tasks. Second, it concerns the way applications are created and deployed to enable such tasks to be performed. Third, it concerns the environment and how it is enhanced by the emergence and ubiquity of new information and functionality.

Some of the new opportunities and challenges that offer the ubiquity for the web include software development based on location, time, network and device independence. [Kap97]

2.3. WIRELESS INTERNET ACCESS

2.3.1. WAP (Wireless Application Protocol)

The proliferation of mobile communications devices has brought about a new technological challenge. How this new devices can be used to enable Internet access?.

To answer this challenge, companies like Nokia, Ericsson, Motorola and Phone (now Openwave Systems), looked for their own solutions for the possibility of delivering contents over wireless networks. These companies created the WapForum - today more than 500 companies belong to this wireless industry association- with the main objective of ensuring product interoperability between its products. [Gue02]

They also generated the Wireless Access Protocol (WAP). WAP can be understood as: “An open, global specification that empowers mobile users with wireless devices

to access easily and interact with information and services instantly"; and, "A protocol that the telephone industry has designed to deliver mobile services to cellular phones in the context of the telephony infrastructure (i.e. phones, bandwidth, scalability)." [Ble01,Gue02,Dor01]

WAP defines:

- A model of standard names for which URIs (Universal/Uniform Resource Identifier) are defined in the World Wide Web in order to identify local resources of the device and the URLs (Universal/Uniform Resource Locator) also defined in the WWW to identify the WAP content in information servers.

- A standard content format, based on the WWW.

- Some standard communication protocols, which allow for the communication between the mobile terminals microbrowser and the on-line Web server.

- The corresponding microbrowser found in the mobile terminal is responsible for coordinating with the gateway, which interprets the WAP and WEB request, which, in turn, is processed and redirected to the appropriate information server.

WAP client applications make a request very similar in concept to the URL concept in use on the Web. A WAP request is routed through a WAP gateway, which acts as an intermediary between the client (GSM, Code Division Multiple Access- CDMA, Time Division Multiple Access-TDMA, etc.) and the computing WAP gateway. The gateway then processes the request, retrieves contents or calls server side scripts, then formats data for return to the client. This data is formatted as WML, a markup language that is an application of Extensible Markup Language (XML) [Nam02]. Once the WML has been prepared (known as a deck), the gateway then sends the completed request back to the client for display or processing. The client retrieves the first card of the deck and displays it on the monitor.

The application level of the WAP protocol denominated WAE (Wireless Application Environment), includes the following functionalities:

- A support for the WML language, similar to HTML but optimised for the inherent limitations found in mobile devices.

- A scripting language, denominated WMLScript, similar in its syntax to JavaScript, which, on the client side, allows for the accomplishment of specific tasks such as validations.

The process that is followed when an available site for WAP is accessed is the following: When a user has a mobile device with access to the Internet, the device has a microbrowser installed in it. When a request is made to a URL that has associated WAP content, this request travels through the wireless network and gets to the server through a gateway, communicating with the application to create a response. [Hil01]

According to WAP1 standard, the following protocols are used between WAP Mobile and WAP gateway: (See Figure 1)

- Wirelesses Session Protocol (WSP) is used for the WAP connection towards WAP mobile (not the same as HTTP sessions). Two modes can be used through a WAP browser connection-oriented mode and connectionless mode. Connectionless mode is based directly on Wireless Datagram Protocol (WDP) and has a lower overhead.

- Wireless Transaction Protocol (WTP) is a lightweight transaction protocol for connection-oriented services.

- Wireless Transport Layer Security (WTLS) handles the communication security like authentication and encryption.

- Wireless Datagram Protocol (WDP) makes the WAP stack adaptation to the common Internet User Datagram Protocol (UDP).

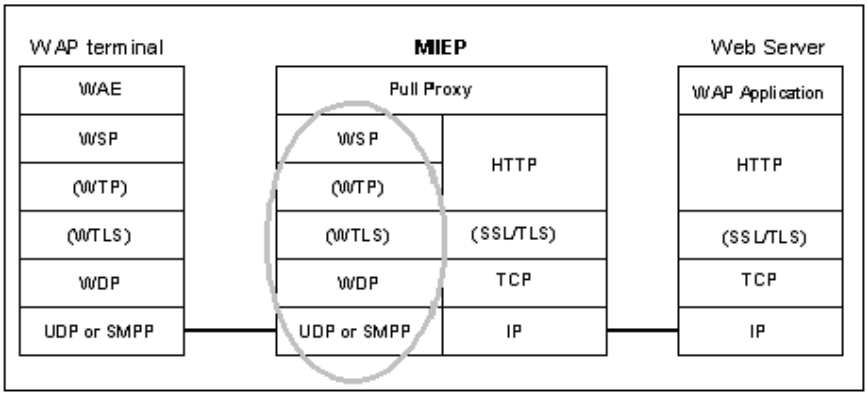


Figure 1. WAP Version1 Protocol Layers vs. Internet Protocol [Eri04]

The most common USED language for the construction of this kind of documents is WML (Wireless Markup Language), which is designed to permit the generation of dynamic content adapted to the limitations of mobile devices. In order to obtain client-side interactions, it is possible to use WMLScript. WMLScript is to WML as Javascript is to HTML. [Dat02]

The difference between the WAP1 and WAP2 stacks is that WAP2 uses Wireless Profiled Hypertext Transfer Protocol (W-HTTP) and TCP to communicate with WAP gateways.

The second generation WAP architecture does not have strict divisions between the server, gateway, and user-terminal environment.

An important feature of WAP2 is the introduction of Internet protocols into the WAP environment, enabling HTTP (Hyper Text Transfer Protocol) over TCP/IP to be used all the way to the wireless device (See Figure 2). This support has been motivated by the emergence of high-speed wireless networks, for example, 2.5G and 3G.

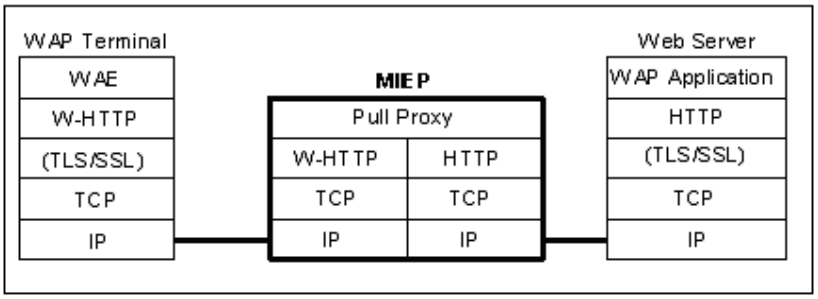


Figure 2. The second-generation WAP architecture [Eri04]

WAP technology does have constraints, most notably device limitations as the following: [Aus00]

- Handsets are small, and typically feature a monochromatic display. This makes it difficult to create an aesthetically pleasing interface for the user (although colour screens have recently been launched). The developers must take into account that users of this technology are used to voice interfaces and they must therefore try to find solutions that will allow an easy access to the applications.

- The devices have several hardware restrictions such as little memory, low power CPU, small or no secondary storage. Therefore interactions with the client must be carefully planned and optimised.

- Also, the limited bandwidth available today, and the low broadcasting speed, means that content has to be limited so it can be sent efficiently. This situation has improved with the new generations of network standards, GPRS (General Packet Radio System) and UMTS (Universal Mobile Telecommunications Systems) that have a intensive use of video and multi-tasking. [And00, Mur01]

- Another constraint is the lack of a "real" standard supported by wireless devices. Each device works with different wireless protocols. For example, some companies use particular extensions of the language, so you can expect

different behaviour of the same code in different devices. Optimisations made for particular devices usually cause unexpected results in other devices.

- Another great weakness of the wireless channels. There is a special difficulty here with the commercial transaction development. Industry initiatives have to include security management with the SSL (Secure Socket Layers) protocols, and they can supply security features for users like authentication, authorisation, and data encryption to address the risks associated with remote access.

- The last major constraint of WAP technology is its high cost, which leads to a very low percentage of mobile devices users that can have access to this service.

2.3.2. I-Mode

I-Mode was launched in February of 1999 [Aus00, Doc00,Doc01,Doc02,Doc03], constituting itself as the first Mobile Internet service in the world in offering a always on (always active) connection and packet switching communication. I-Mode has grown from 11 million in August of 2000 to more than 40 million in April of the 2004. During this time there have been appeared a great number of applications as an added value of the service users. [Mur01,Rat00]

One of the main I-Mode characteristics is the complete separation between the voice chanels and the rest of the information, therefore avoiding the congestion of the circuits of voice.

From the users point of view, the service was offered on the basis of a quite low monthly fee which allowed the downloading of a sufficient amount of information. Currently millions of Japanese are having their first experience of the Internet from their mobile devices. Unlike WAP, this technology was not offered initially as an access to Internet from the mobile phones, but rather like an additional service to the already existing ones. The success of this technology has been helped by the high cost

of the access to Internet from PCs in Japan (approximately 67 US dollars for 20 hours) [Wal02]

I-Mode is equipped with a system of micro payments which offers the possibility of making small payments by connection to the banking system. The paymets appear in the monthly account of the user, eliminating the necessity of the credit card, and in this way becoming the greatest success experience in mobile electronic commerce (m-commerce). I-Mode was first: [Rat00]

- To allow connection always-on to the user and therefore invoicing at low cost by the amount of downloaded information, under a high speed scheme.

- To count on an infrastructure of third generation network.

- To allow the use of colours in the devices at the beginning of year 2001, as well as first in allowing images and the downloading of Java programs to devices.

- To develop a functional solution of mobile electronic commerce (m-commerce) based on its system of micro payments.

The layers of the I-Mode protocol can be seen in the Figure 3.

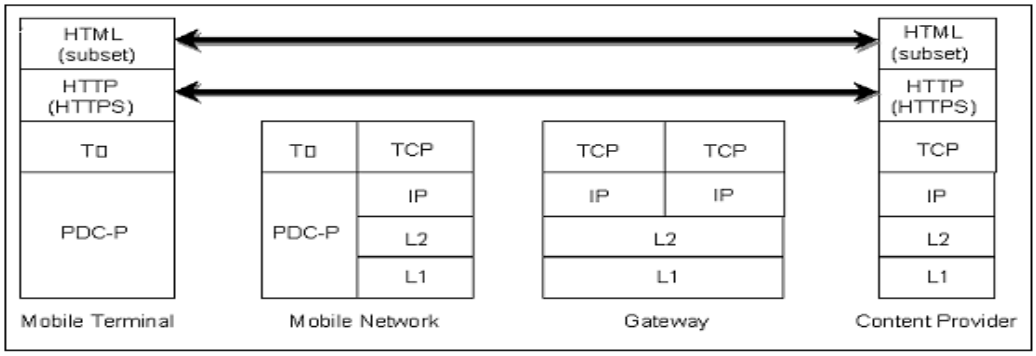


Figure 3. I-Mode Layers [Doc01]

The I-Mode network built by DoComo defines the complete service envelope, including the server and client, the markup languages, Java functionality, the gateway

architecture, transport layer and other layer protocols, and session requirements (similar to WAP). I-Mode comprises a multilayer delivery model, and, although the only working example so far at the beginning has been fielded over DoCoMo'S bearer network, the I-Mode model could be ported to any 2G or upper network.

All the previously mentioned advantages have as departure point the system used in the network layer of both protocols. I-Mode was based from the beginning in commutation of packages and WAP on GSM networks based on circuit switching.

The difference in the technologies of the physical layer of the protocols had a direct incidence in the cost of the service facing the users and therefore in the acceptance of them. I-Mode was based on the collection of flat tariffs and later in collections by amount of information downloaded, as well as providing pleasant user interfaces.

2.3.3 Access from PDAs

A PDA or Personal Digital Assistant, is a mobile device that can integrate the services of a digital agenda, computer, address book, task and notepad, connection to Internet, sending and receiving electronic mails, telephone, and camera, among others. They can be considered as an extension of the desktop computer.

The operating systems with which they work are simple, intuitive and have the capacity to keep much information in little space PDAs come in two main groups: those using the Palm OS and those using the Windows CE, Each class has some distinct pros and cons.

Other environment are summarized in Table 1). Neutrino from QNX is very well suited for car devices and BeOS is used for set-top boxes and other multimedia devices.

	Palm	EPOC	Windows CE	QNX	BeOS	Embedded Linux
Supported processors	Motorola Dragon Ball	NEC, ARM	x86, SH3/4, PowerPC MIPS	x86, PowerPC, MIPS	x86, PowerPC	x86, Power PC, MIPS, ARM
Operating system structure	Layered	Layered	Modular	Modular	Layered	Modular
Memory protection	No	Yes	Yes	Yes	Yes	Yes
Operating system size	Tiny	Small	Small	Medium	Large	Small-Large
Security	None	Low	High	Medium	High	High
Multi-tasking	No	Yes	Yes	Yes	Yes	Yes
Examples of pervasive devices	PDAs	Mobile Phones	PocketPC	Car devices, Internet Appliances	Set-top boxes	Set-top boxes

Table 1. Features of different operating systems for mobile devices

PDAs run on batteries; some are rechargeable (Palm Vx and Handspring Prism) and others are disposable (Palm VII and Handspring Neo). Disposable batteries last approximately 2 weeks (longer on non-colour models), even with heavy usage.

It is possible to add to them peripheral memory, applications, multimedia content and accessories like keyboards, modems, GPRS and GPS (Global Positioning System) modules, and cameras.

Unlike portable computers, most PDAs began as pen-based, using a stylus rather than a keyboard for input. This means that they also incorporated handwriting recognition features. Some PDAs can also react to voice input by using voice recognition technologies. Today PDAs are available in either a stylus or keyboard version.

Possibly the most important limitation is the low capacity of on-screen unfolding of information in these devices, so it is necessary to develop specific applications for specific PDA models; and it is possible that a designed application that runs on a PDA model, surely could run on other models.

The PDA can be connected to the Web through different mechanisms like: (See Figure 4)

- Using a wireless connection to a Wireless LAN. In this way is also possible to connect to a corporate network (the initial tests in this work were performed way). In this case the greater disadvantage is the coverage of the radio network if further mobility is needed.

- By means of a GPRS module included in the PDA. This facilitates mobility, but is limited to the places where there is GPRS access

- Using a Bluetooth connection (technology that facilitates the connection without cables between devices) with a GPRS mobile phone acting as wireless modem. [Gil01]

- By means of a connection with a computer through the serial port and using synchronization software like ActiveSync. In this way, the information and programs stored in the PDA are synchronized with the computer via a special cradle that you connect to your desktop.

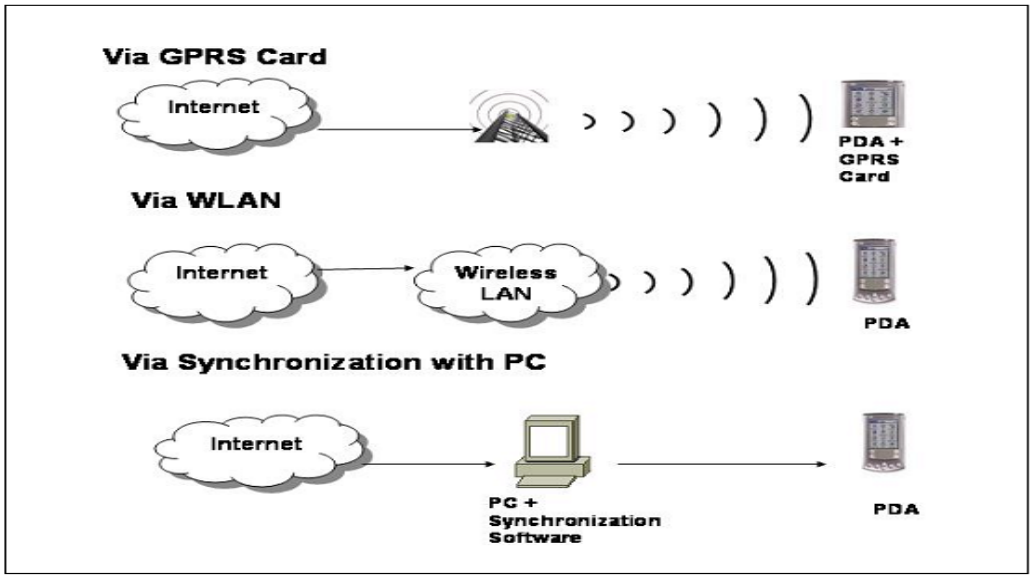


Figure 4. Access to web applications from a PDA.

2.4. VOICE PORTALS

A voice portal uses voice synthesis, voice recognition, and pre-recorded messages to guide the user through the content of the portal. It permits surfing the web using voice commands and the service can be accessed speaking into ordinary (or mobile) phones. In addition, DTMF (Dual Tone Multi-Frequency) tones can be sent from the user's phone and will be detected by the server. [Cer03, Gam03]

Figure 5 shows the elements involved in an interaction system with Web content through voice, paying attention to the interaction in a traditional telephone system: [Sha02]

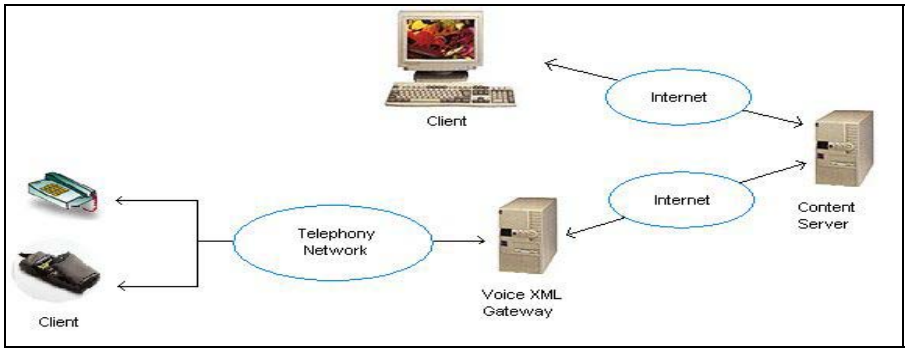


Figure 5. Architecture of a Web System Based on Voice

The user that connects to the Internet through a telephone dials a number and with the telephone network, s/he connects to a gateway. Such gateway accepts the call (for that purpose, there is a telephone interface connected to an E1 connection, which is a high capacity line) and makes a call with the HTTP protocol to a content server that gives the source document.

The content server can make queries to databases to obtain updated information [Ker02]. In order to exchange data between the gateway and the server, the VoiceXML language is used, which is a markup language oriented to communication through voice. [Edg01]

In case a user gets connected through a computer, the navigator is in the client's part. The computer should have a navigator that has the modules previously mentioned. In this way, the user requests a page through the Internet connection, s/he receives it and the navigator interprets it and starts the functioning of the identification services and the voice synthesis to interact with the user. [Gar00]

The Gateway

In order to interact with the system, a browser is necessary. In the case of telephones, the browser is included in the gateway while, if accessing with a computer, the browser is an application that is locally included in the client. A browser needs to have at least three types of modules implemented:

- VoiceXML Parser: It is in charge of interpreting the VoiceXML document.
- Voice Identification (Automatic Speech Recognition): It is in charge of identifying what the user says, taking into account the vocabulary and grammar sets that contain the words or phrases accepted.
- Voice synthesis (TTS:Text-to-Speech-TTS): The system reproduces the text with voice.

This structure allows the use of "light" clients, that is, those that do not have to carry out much information processing because the recognition and synthesis tasks are implemented in the intermediate platform. VoiceXML is oriented to this type of clients.

There are free gateways, which are especially interesting for the development of prototypes of voice applications. There are companies that offer a free gateway service. It is the case of VoiceGenie, Bevocal, HeyAnita, Tellme, or Voxpilot. Using the free gateway is useful to carry out trials. Some of the advantages and disadvantages are:

It is also possible to contract the gateway which gives the user more possibilities. With this service type, it is possible to answer a long number of calls due to the fact that several ports can be arranged for simultaneous call attention.

2.5. MARKUP AND CONTENT/PRESENTATION SEPARATION

In this section, markup languages used with mobile devices are describe in more detail.

As previously discussed, one of the great differences in the Web access from mobile devices is the markup language they support. The markup is a term that is applied to any set of codes or labels that are added to a data set, with the purpose of giving information related to their meaning or format. [Jon01]

Web documents use tags or labels that help to describe their content and format. This concept has been used previously in other non-web contexts, such as RTF: (Rich Text Format), a way to store information related to the format such as bold or italics.

Existing markup languages are based on SGML (Standard Generalized Markup Language, ISO 8879), that dates from 1986. It began to be developed from principles of 70's, and at that time it was based on GML (Generalized Markup Language) created by IBM in 1969.

SGML provides a consistent and precise way to apply tags to describe the parts that compose documents, allowing document interchange between different platforms. Nevertheless, SGML is commonly said to be too difficult.

Then, trying to maintain the idea of SGML, XML was derived as a simplified subset of SGML, eliminating the more troublesome and less useful parts. XML is a metalanguage: it is a language to define languages. XML it a language for organizing – not merely presenting – data. XML is more flexible than a fixed-format markup language and lets you define your own custom tags to represent data logically. XML adds content and gives basic meaning to data[Jai01]

Also with SGML as basis, HTML was defined. It has to be stated that HTML is simple a language, whereas XML is metalanguage [Sel02, Mar01].

XML was created in some respects as a means to bring order to the chaos that HTML had created. The first major benefit of using XML for storing Web content was that it

enforces a clean separation of data or content from presentation or formatting. Separating the concepts content, presentation, and interaction allows more easily composable specifications. [2] [Ber01b, Sel02]

The evolution of markup languages used with mobile devices it's shown in Figure 6.

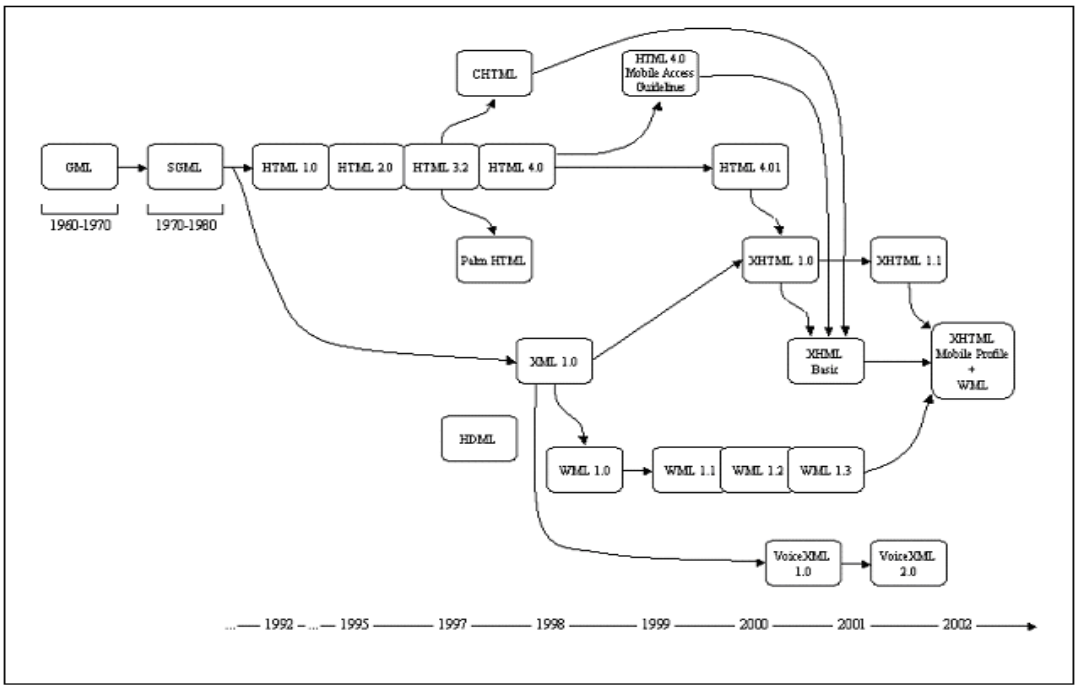


Figure 6. Evolution of the Markup Languages [Son02]

Mobile WAP phones traditionally have used WML, while other devices have used variant of HTML. Recently WML and HTML have united their forces in the recent specification WAP 2,0 that uses Extensible HyperText Markup Language-XHTML) Mobile Profile like language for content WAP. There is great probability that all new devices support XHTML Mobile Profile in the future. [Med01]

Japan has three competing standards -HDML, Docomo's Compact HTML (cHTML) and J-Phone's Mobile Markup Language(MML) - none of which is compatible with WML as it's used in North-Amercian and European systems. [Wal02]

[2] By moving to this architecture, it would also become easier to separate the disciplines of database integration from graphic layout and presentation.

Next a review of the main markup languages involved in the development of ubiquitous web applications Web is presented

- *Hypertext Markup Language(HTML)*

HTML is the de facto standard uses on the Web. The current version, 4.01, is a successor to version 4.0, 3.2 and 1.0. The language was designed for desktop devices, but new devices have included support for different versions of HTML as well. Because of the limitations of mobile devices (small display space, limited capabilities, input methods, etc.) all HTML features are not appropriate. A Pocket PC 2000 powered PDA can handle HTML version 3.2. The new Pocket PC 2002 understands more advanced HTML features. The Nokia 9210i Communicator has HTML 4.01 support and it understands frames and cascading style sheets. HTML browsers for mobile Java enabled devices are also available.

- *Handled Device Markup Language (HDML)*

HDML is Openwave's proprietary language which can only be viewed on mobile phones that use the Openwave browser. It was created before WML, it uses Handled Device Transport Protocol (HDTP) instead of WAP and it is mainly used in North America and Japan. HDML is the only language that older phones understand, but it may be on its way to obsolescence since all new phones support fully XML-compliant WML.

- *Compact HTML (CHTML)*

CHTML, is a subset of HTML targeted for small appliances such as smart phones, communicators and mobile PDAs. The language is defined so that all basic operations can be performed using a combination of four buttons. Scrolling is not supported, because it is assumed that properly designed pages will fit on a single screen. CHTML includes support for GIF and JPEG images, but tables, image maps, multiple character fonts and styles, background colour and images, frames and style sheets are excluded. [W3C02a, W3C98]

The CHTML specification can be thought of as a stripped-down version of HTML with the new attribute “accesskey “ that can be used in anchor and input tags. CHTML does not use HTML 4.0 positions as layers of style sheets and JavaScript is not supported.

This language is also known like IHTML (I-Mode HTML) especially from the expansion of the I-Mode technology outside Japan.

- *Palm HTML*

Palm HTML is the markup language used on Palm PDAs, is based on the HTML 3.2 specification, but the following 3.2 features are not supported: cascading style sheets, frames, image maps, animated GIFs, Java applets, JavaScript, layers, nested tables, multiple font faces, and the <vspace>, <sub>, <link>, <isindex> tags. On the other hand Palm HTML includes several new tags, variables and meta tags not found in HTML specification.

-*WML*

WML is an XML document type defined by a standard XML Document Type Definition. WML is based on deck/cards paradigm.

The deck of cards metaphor is designed to take advantage of small display areas on handled devices. Instead of continually requesting and retrieving cards (the WAP equivalent of HTML pages), each client request results in retrieval of a desk of one or more cards. The client device can also use the logic of embedded WML Script (the WAP equivalent of client-side JavaScript).

The purpose of WMLScript is to provide additional intelligence through client side procedural logic. It is based on ECMAScript (which is based on JavaScript), modified to support low-bandwidth communications and thin client.

-*XHTML Basic*

XHTML is a family of current and future document types and modules that reproduce, subset, and extend HTML 4. XHTML is a subset of XHTML 1.0 Strict as well as XHTML 1.1, built from XHTML modules.

XHTML family document types are XML based, and ultimately are designed to work in conjunction with XML-based user agents. XHTML Basic provides us with a document type that can be shared across various devices (desktops, TV, or mobile phones). It was designed for Web clients that do not support the full set of XHTML features; a common base that may be extended.

-*XHTML Mobile Profile*

XHTML Mobile Profile. XML-based markup language that contains XHTML Basic and a few additional elements from full XHTML 1.1.

-*VoiceXML*

VoiceXML is an XML based language that allows the design of services based on voice for Internet navigation purposes.

Different companies have created independent versions of phone markup languages in the mid 90's. AT&T&Bell Labs created PML (Phone Markup Language), Motorola developed VoxML and IBM SpeechML Other experimental languages for voice browsers are TalkML developed by Hewlett Packard and CallML by Voxeo.

Later the VoiceXML Forum was created, and today it has over 500 members. It defines the framework of an applicable voice markup language standard, VoiceXML.

The VoiceXML infrastructure present in the gateway includes processes such as voice-recognition and text-to-speech conversion. VoiceXML is only used to describe the conversation between the voice portal and the caller (the user interface of the voice application).

The technology for speaker independent voice recognition (Speech-to-text conversion) is based on grammars that specify a set of possible strings to match. In VoiceXML, acceptable speech input is specified using a speech grammar, which provides a template to describe valid user input.

VoiceXML 1.0 does not define a standard for speech grammars, it introduces the Speech Recognition Grammar Format (SRGF), for defining speech grammars. The SRGF grammar is fully embedded within the VoiceXML. The only grammar that is mandatory for VoiceXML 2.0 compliant browsers is the XML form of SRGF.

VoiceXML also includes a full-embedded XML language for text-to-speech markup, The Speech Synthesis Markup Language.

In 2000, VoiceXML 1.0 was accepted by the W3C as the basic for developing dialog-based markup language. In 2002, VoiceXML 2.0 became an official W3C recommendation. The W3C's Voice Browser Working Group released the First Public Working Draft of the Voice Extensible Markup Language (VoiceXML) 2.1 on March, 2004, which is Fully backwards-compatible with VoiceXML 2.0. The draft's purpose is to standardize eight additional features implemented by VoiceXML platforms.

2.6. DEFINITIONS OF CONTEXT

Key attributes of the ubiquitous computing concept are the mobility, interconnectivity and context-awareness of these devices. [Pas99]

Mobility allows the devices to be taken with the user if necessary, or moved to suitable places, etc. Interconnectivity allows the devices to be aware of each other and to exchange information or even control each other. Context-awareness allows the device to adapt its behaviour to the circumstances in which it finds itself.

Many researchers have attempted to define context by enumerating examples of contexts. Schilit divides context into three categories: [Sch94]

- Computing context, such as network connectivity, communication costs, and communication bandwidth, and nearby resources such as printers, displays, and workstations.

- User context, such as the user's profile, location, people nearby, even the current social situation.

- Physical context, such as lighting, noise levels, traffic conditions, and temperature.

In this sense we can include the "Time context", such as time of a day, week, month, and season of the year, and then we can obtain a context history, which could also be useful for certain applications.

Some other researchers try to formally define context:

- Schmidt et al. define context as "knowledge about the user's and IT device's state, including surroundings, situation, and to a less extent, location" [Sat99].

- Dey defines context as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" [Dey99].

- "Primary" contexts, including location, entity, activity and time, act as indices into other sources of contextual information [Dey99]. For example, knowing the current location and current time, together with the user's calendar, the application will have a good idea of the user's current social situation, such as having a meeting, sitting in the class, waiting in the airport, and so on.

- Context is the set of environmental states and settings that either determines an application's behaviour or in which an application event occurs and is interesting to the user. [Che00]

Context is becoming increasingly important in ubiquitous computing, where the user's context often changes. Ubiquitous computing applications need a constant flow of information about their environment to be able to adapt to their changing context. The context includes circumstances of consumptions relevant for a ubiquitous web application, mainly dealing with the questions why, when, who, where, when. Supporting the right abstractions and services for handling context makes it easier to design, build and evolve context-aware applications.

2.7. CONCLUSIONS

In this chapter the main protocols that allow the wireless access to Internet have been studied, which are independent of the network technology available in the physical layer and which at this time are found to be used under 2.5 G or third generation networks, in which the available bandwidth allows developments of richer contents.

The I-Mode protocol was created with a technology that was totally dependent on its manufacturer (Docomo) and the markup language used by its terminals, CHTML has not been approved within the W3C standards. Elsewhere, the WAP protocol at the beginning used the WML markup language based on the desk/card paradigm, but starting from the 2.0 versions it has been proposed that the new terminals use the XHTML standard for content generation. However, XHTML does not allow backward compatibility, and the navigators included in the old terminals do not have the capacity of visualizing contents based on XHTML nor CSS.

Since the beginning the I-Mode network makes use of existing protocols that can be quickly understood by any developer and its services differ little from those offered on WWW sites. However WAP 2.0 use new standardised protocols based on W3C proposals. Finally, from a strict technical point of view, WAP might be the superior solution.

In relation to the voice portals, its main components and typical markup language accepted by W3C and VoiceXML have been described.

Starting from the voice markup evolution analysis, it can be seen how the tendency is to unify all markup towards the convergence based on XHTML. However, the markup difference is not the only problem to be treated in the content generation that may adapt to multiple devices, since many other properties exist that must be analysed.

The voice applications inherit the advantages and disadvantages of technologies like the voice synthesis and voice recognition. The quality of the recognition, the background noises and the difficulty for the user to pay constant attention must be specially considered. Also in the process of text-to-speech conversion it is important to simulate human-to-human conversation as best as possible.

Finally it can be noticed how the context concept has been addressed by diverse scientific communities. Starting from the context analysis, it is possible to approach a level of abstraction associated to the environment on which a ubiquitous application can be carried out. Some of the aspects included in its treatment are user location, the device used to access contents, as well as the possible variation in time of these characteristics as well as user preferences.

3. TOWARDS MODELLING AND DEVELOPMENT OF UBIQUITOUS WEB APPLICATIONS

3.1 SUMMARY

In this chapter the existing proposals for modelling of ubiquitous Web applications are described, as well as existing solutions for independent device content generation.

To obtain this objective, in the first part of the chapter the founding concepts that come from diverse areas of computational sciences are presented, which serve as base for the developments and proposals generated by the academic and scientific communities.

To achieve this, two perspectives are analysed: that of the Software Engineering, and that of the Knowledge Representation which in turn forms part of Artificial Intelligence.

In relation to the Software Engineering, an outline which shows the evolution of this type of technique is presented, from desktop application modelling to Web application modelling, some of which incorporate elements that can be used in ubiquitous application modelling as well. Emphasis is placed on the differentiating elements for Web application modelling, and the actual state is proposed in relation to the development for ubiquitous applications.

The approximation based on Artificial Intelligence and specifically on Knowledge Representation describes models that have been developed to represent information, such as semantic networks, logical representations, and the definition of vocabularies by means of ontologies.

The former concepts are the founding base for the development of the FIPA (Foundation for Intelligent Physical Agents) ontology for device representation and models proposed by the W3C consortium, such as RDF for describing resources, and CC/PP (Composite Capabilities/Preferences Profile) as guide frame for the development of vocabularies for device description. In the same way, the initiative of semantics web that looks to generate mechanisms that allow an "intelligent"

information exchange is described, which goes beyond labelling and the separation of content/presentation obtained through standards such as XML. [Mel01]

Subsequently, the device independence principles that have also been generated by W3C are presented, in order to guide application developments that function at any given moment, anywhere and by any means. Once the principles are proposed, general techniques are described that can be used for the generation of content in multiple formats, based mainly on transformations carried out in different layers of the Web information exchange process (client, gateway and server).

Finally, tangible solutions that are available in the form of libraries or commercial products are described, that try to propose ways of obtaining application development that can be run in multiple devices.

3.2. INTRODUCTION

Some of the considerations associated to the problems of the Ubiquitous web applications are the following:

- The diversity between the existing connectivity of the protocols, for instance WAP and I-Mode, which involve different layers and have different associated models, with which, it is difficult to develop a unified model.

- Another existing difficulty in the development of mobile applications is their testing process, because the acquisition of each one of the target devices to verify the functionality of the application is not easy. It is true that we have emulators (software that acts in a similar way than the available browsers in the devices), but often there are differences between the operation in the emulator and the real working order in the device.

- The diversity of target markup languages for the existing micro-browser in each device and also, the coexistence of more than one markup language. This is how different models of WAP devices support different versions of

WML or XHTML. If the access is based on I-Mode technology, it will support some version of CHTML as a markup language, and if it is a PDA, it is possible that it supports some simplified version of HTML.

Additionally, the current WAP recommendations are based on a variation of XHTML Mobile Profile and the combined use of the XHTML and CSS (Cascade Style Sheets). Only the new devices that appear in the market, give some support to this language, whereas may existing devices does not allow you to visualize this type of contents.

- The previous difficulty is related to characteristics of the device like the number of lines by screen, its resolution and also, the maximum size that can be received, for instance, 5KB in some I-Mode devices, or 2KB in the first WAP terminals.

- The existing physical layer available in wireless networks where the device gains access to the database is important because of the evolutions of network generations; it allows you to have a better connectivity and bandwidth, for example, a telephone based on WAP technology that gains access to a UMTS network, can have the capacity of deployment of Multimedia contents whereas, this is not possible in a phone based in a GSM network.

- In relation to the presentation of graphic information, in the case of finding a device that does not support a determined images format, the required techniques go further than a simple visualization of a text that indicates the incapacity of deployment. Otherwise, we search to develop content that make an accurate use of the capacities of visualization and the available space on the screen.

- On the other hand, the dynamism that requires the applications make the process of creating contexts difficult, due to, for instance, the obtained answers of a request to a server, which have a varied number of characters, with which the characterization of the required screen sizes is difficult. (For

example; if the obtained city as answer to a query, is Bogotá, or if the obtained answer was Barranquilla.

- Additionally, the navigation structure between the pages can vary notoriously between the diverse devices. For instance, if it displays some content in a PDA, is possible to use lists to enable the selection of one of the elements of the group. However, when this same interface is developed in a mobile phone, it is possible than in some models, the possibility of including a displayed list does not exist, and you have to subdivide the request into more than one page.

Next, the theatrical elements and main involved technologies are described in the way of dealing with the modelling of ubiquitous web applications.

3.3. SOFTWARE ENGINEERING PERPECTIVE

In this section, there are some descriptions about the efforts related to the modelling of the Web applications that at the beginning, could be considered as a feasible approximation to developing the model of Ubiquitous applications.

At the beginning of the 90's, the "desktop" personal computer paradigm predominated. Some considerations to be taken into account included the ways to adapt the application presentations to different screen resolutions. In another area, an application installation process was required, and in many instances an instruction phase for users was included.

Afterwards, with the advent of application access through the Web, new considerations were to be taken into account. On one hand, the application not only had to consider screen resolution, but also aspects such as the browser being employed by the user.

Additionally the application had to be designed to be accessed by any user without previous instruction. Likewise, the interaction is characterized by the use of links based on the hypertext concept.

The requirement for modelling Data Web Applications has been characterized [SchO1] by means of three orthogonal dimensions, comprising levels, aspects and phases.

The levels include content, hypertext and presentation. The content level refers to domain-dependent data. The hypertext level denotes the logical composition of Web pages and navigation structure. The presentation level is concerned with the representation of the hypertext level, e.g., the layout of each page and user interaction.

The aspects comprise the structure and behaviour. Looking at the hypertext level, they include the modelling of page compositions and navigational relationships within, as well as behaviour like computing the endpoint of a certain link at runtime.

The phases include the different phases of the software cycle. They include the analysis, the abstract representation (conceptual modelling), technology-independent design (logical modelling) and technology-dependent design (physical modelling).

On one hand, there are approximations based on the concept of hypertext associated to the documents, and the logic and structural composition of the pages. Within the proposals of the modelling web applications, we can emphasize the following:

- HDM (Hypermedia Design Method) [Gar03] is a proposal generated by the hypermedia community, characterized by the identification of different categories of links, with different rules of representation and the possibility of integrating the structure of an application of hypertext with its semantic navigation, the definition of significant links arise from the conceptual descriptions of designed level.

- RMM (Relationship Management Methodology) [Isa01] uses the E-R diagrams that come from the database modelling to define its model of data, called Relationship Management Data Model (RMDM), which provides a language to describe information objects and mechanisms for navigation in

hypermedia applications. RMM specifies a process of development where it includes stages of analysis of the requirements and modelling of content by means of E-R diagrams.

- The Araneus Data Model (ADM) [Mec99] provides a basis for compact description of large sites. It considers each Web page as an object with identifier (the uniform resource identifier) and a set of attributes. ADM uses the notions of a page schema (which resembles a relation schema in relational databases or a class in object-oriented databases) to model sets of homogeneous pages.

Araneus emphasizes mainly on content and the hypertext. To the process of proposed modelling, they start with a conceptual design of databases, which later comes from a conceptual model. In the same way of RMM, it focuses its proposal of modelling content on the E-R model, on which it is based to define the Araneus Data Model (ADM). The hypertext model is mapped in the content level, using a convention that constitutes a base to the automatic generation of pages.

- OOHDM (Object – Oriented Hypermedia Design Method) [Ros97] uses abstraction and composition mechanisms based on the object oriented modelling techniques. On the other hand, it allows you to make brief descriptions of complex information elements, likewise the specification of patterns of complex navigation and interface transformations.

On the other hand, there are modifications to existing approximations such as the extensions to UML (Unified Modelling Language) developed by Canallen [Con00], by means of which, you may take into account some characteristic elements of the web applications, like the scripts of the client or server which are modelled as UML classes.

Additional efforts like WebML and OOH (object oriented hypermedia) allow you to include context elements, and they are evolving to the generation of trade solutions.

- The WebML[Cer02] is focused on the visual notation to specify the composition characteristics and navigation of hypertext applications. From the WebML proposal, it has originated a tool case called Web Ratio, which generates functional applications automatically from the WebML diagrams. Web ML is also based on E-R and UML models.

  WebML provides modelling abstractions that a computer-aided software engineering (CASE) environment can translate into concrete page templates. According to the WebML approach, Web applications have two orthogonal conceptual dimensions: the data model and the hypertext model.

- OOH [Gom01] is a generic method that suggests a necessary semantic notation to develop web interfaces, indicating how to connect them to logic modules of existing application to facilitate the migration of applications. Using this method as a starting point has generated a tool CASE that makes the modelling of web applications easier.

The interaction of the user with mobile devices consists of two functions: presentation of data and input of data. Presentations break down into navigation and style.

The input models of devices can be abstracted to a textual input format, regardless of whether input is received from a keyboard or a keypad, voice recognition, character recognition (for example, pen-based input), or other means.

A special problem which can be abstracted to a higher level, is presented by devices that use shortcut keys, macro keys, softkeys, or other devices to represent a textual input as a shortcut. Menus can be seen as a special case of such devices.

The navigation is the other aspect of the presentation and the hardest to abstract from the device. In devices with very small screens, as well as in voice-menu systems, alternative paradigms are emerging for navigation of content, i.e., the "deck of card" used in WML.

Modelling all possible future presentation format is not feasible, however. There are two possible ways to meet this goal: either restrict presentation to a single set of devices or model the navigation of the content in a device-independent way.

Modelling the navigation in a device-independent way also lets us retain control over how it will be navigated in different devices. Navigation is actually a larger part of the user experience that the presentation, although these tend to be mixed.

On the other hand, some users tend to ignore style information, frequently switching off images to enhance transmission speed or due to economical reasons.

To change the navigational paradigm for a document, however, it is not enough to change the formatting of the content. In the case of a small-screen device, the information will need to be filtered, if the user is to avoid handling enormous amounts of irrelevant information. Then, a document that is device independent must contain all possible navigational models or a subset of all possible navigational models for which the content is enabled.

In addition, an application in today's world should still work with devices that come out months and perhaps even years later. The key to this functionality is to follow the standards as much as possible.

Finally, the link between the logical and conceptual modelling, and their physical modelling of the applications is not clear in the analysed proposals, neither is the way in which the context aspects related to the web access from the devices, can be specified. On the other hand, there are not proposals for the treatment of the context context and navigation differences that can be present in different types of devices that can gain access simultaneously to a web application.

3.4. ARTIFICIAL INTELLIGENCE PERSPECTIVE: KNOWLEDGE ACQUISITION AND REPRESENTATION

AI researchers have evolved into two major communities: [Fen01]

- Knowledge acquisition and engineering, which deals with acquiring and modelling knowledge (the human-oriented problem) and

- Knowledge representation, which deals with representing knowledge and reasoning about it (the computer oriented problem).

However they have worked with big limitations. Knowledge acquisition is too costly, and the knowledge representation systems were created with mainly isolated, brittle, and small solutions for minor problems in the past.

With the web this situation has changed drastically. It is very difficult to process something in the web due to, the volume of information that it contains, it is not possible to handle operations by hand. There is a huge quantity of information and knowledge on the current web, furthermore is not well organized.

Knowledge Management is also concerned with acquiring, maintaining, and accessing an organization's knowledge. Its purpose is to exploit an organization's intellectual assets for greater productivity, new values, and increased competitiveness. [Coo00]

3.4.1 Knowledge Acquisition

Knowledge Acquisition and Machine Learning formalisms and methods include among others:

- Bayesian Classifiers
- Bayesian Networks
- Finite State Machines and Hidden Markov Models
- Artificial Neural Networks
- Relational Learning

3.4.2 Knowledge Representation

The devices seen as elements available in the Web requires some kind of representation mechanism. A knowledge representation is a class of data structure that uses symbols to represent objects, concepts, and processes.

Two of the classic mechanisms of the knowledge representation are the semantic networks and the logic representations. [Kap97]

3.4.2.1. Semantic Networks and Logic Representations

Semantic networks are mechanisms from which many other structures of knowledge have their origin. The semantic networks can be used to represent different types of phenomena.

There are different ways in which the semantic networks can be represented. Nevertheless one of most common is the representation based on graphs - nodes connected by arcs. Conceptually, the nodes represent entities such as concepts, objects, and processes, and the arcs represent relations among them. An example of a representation of a device based on this mechanism is shown in the Figure 7.

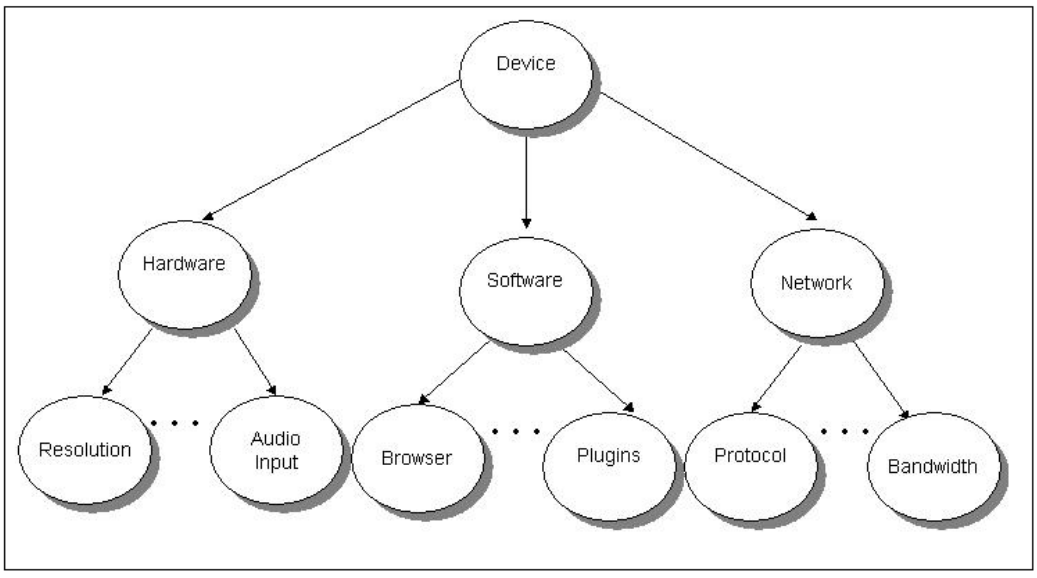


Figure 7. Semantic network representing a device

The logic representations of knowledge are other classic way of representation. Some benefits related to the use logic representations include the strength of their mathematical foundation and the existence of a computational structure that widely supports this mechanism, based mainly on languages like Prolog. An example of a logical representation is the following one:

Owner (Eduardo,PDA)

This representation could indicate that Eduardo is the owner of the PDA

3.4.2.2. Vocabularies and Ontologies

A vocabulary is analogous to a dictionary. It identifies all the possible attributes in a schema, which is similar to a database schema.

The ontologies list the words and the relationships between them that allow the communication in on understandable way by machines. Databases use ontologies during the design of their entity-relationship diagrams (implicitly) and represent the nodes and their schemes

Ontologies try to capture knowledge about the world, and the question naturally arises as to how this knowledge should be represented.

One natural approach is to divide the world into classes of objects with common properties, identifying with common properties, inheriting all the properties of the more general class and (possibly) adding new properties of their own. [Mae01, Sow99]

There are many definitions of ontologies, some of them include:

- An ontology is an explicit specification of the concepts in a domain and the relations among them, which provides a formal vocabulary for information exchange. [Gru93]

- Ontologies are structured vocabularies shared by communities of users. Therefore, they are simple: tress of terms, or taxonomies of classes described by their names and their attributes.

- An ontology is a formal, explicit specification of a shared conceptualisation. [Gru93]
- Ontologies serve as metadata schemas, providing a controlled vocabulary of concepts, each with explicitly defined and machine processable semantics. By defining shared and common domain theories, ontologies help people and machines to communicate concisely - supporting semantics exchange, not just syntax.

Complex mapping and reasoning about those mapping are necessary for a comparing and combining ontologies and for integrating data or services described using different ontologies.

Some of the main functions of the ontologies are:

- To define words used in the construction of systems to facilitate their correct construction and later understanding.

- To define high level abstractions necessary to communicate in major contexts.

- To share the cost of the acquisition and the maintenance of the knowledge.

3.5. EFFORTS OF INTERNATIONAL COMMUNITIES

3.5.1 FIPA Ontology for Device Description

The FIPA (Foundation for Intelligent Physical Agents) has proposed an ontology for device representation that has served as the base for other developments. [FIP02]

A description of device capabilities and user preferences is known as a profile. This profile is sent by the client to the server to guide the creation, adaptation or selection of content presented to the device.

Provided that two devices D1 and D2 have a connection, they may exchange device profiles (either directly or through a brokering agency) and acquire a list of services provided by the other device.

The list of services may include both hardware and software services, for example: a software component that provides access to a hardware component of the device. The profile needs to support the identification of services for various input and output capabilities, such as audio input and output.

The FIPA-device ontology can be used by agents when communicating about devices. Agents pass profiles of devices to each other and validate them against the FIPA-device ontology. The profiles come in handy for example in a situation where memory- or processing-intensive actions take place; agent A1 can ask agent A2 whether device D has enough capabilities to handle some task A1 has in mind.

3.5.2. World Wide Web Consortium Proposals

This World Wide Web Consortium – W3C was founded in 1994, by Tim Berners-Lee with encouragement and support from the Massachusetts Institute of Technology (MIT) Laboratory for Computer Science (LCS).

The purpose of the World Wide Web Consortium (W3C) is “to lead the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability.”

3.5.2.1. Metadata, Resources, RDF and RDFS

Metadata is data about data. Specifically, the term refers to data used to identify, describe, or locate information resources (any object with a Uniform Resource

Identifier-URI as its address, i.e. WWW pages or parts of web pages, mobile devices), whether these resources are physical or electronic[3]. [Mil98]

While structured metadata processed by computers is relatively new, the basic concept of metadata has been used for many years in helping manage and use large collections of information (for example, a catalogue of a library is a metadata because it describes publications).

The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web proposed by W3C. [Kle01,W3C04b,W3C04c]

RDF allows the specification of the semantics of data in a standardised, interoperable manner. It also provides mechanisms to explicitly represent services, processes, and business models, while allowing recognition of no explicit information.

The RDF data model is equivalent to the semantic networks formalism. It consists of three object types: resources are described by RDF expressions and are always named by URIs plus optional anchor Ids. A property is a specific aspect, characteristic, attribute, or relation used to describe a resource. A statement specific resource together with a named property plus the value of that property for that resource.

RDF allows descriptions of Web resources to be made available in machine understandable form. This enables the semantics of objects to be expressible and exploitable. Once widely deployed, this will enable services to develop processing rules for automated decision-making about Web resources.

RDF is based on a concrete formal model utilising directed graphs that includes the semantics of resource description. The basic concept is that a Resource is described through a collection of Properties called an RDF Description. Each of these properties has a property type and value.

[3] According to the present web, a resource is anything that can exist physically in the web like a web page, or a part of it, an image, etc

Any resource can be described with RDF as long as the resource is identifiable with a URI as shown in Figure 8,9.

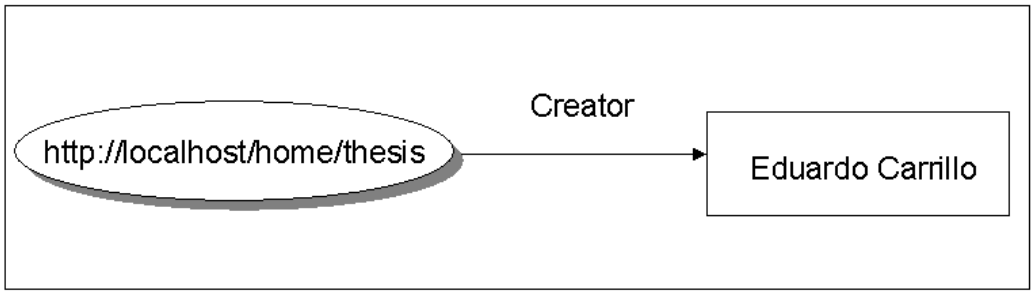


Figure 8. Example RDF Graph

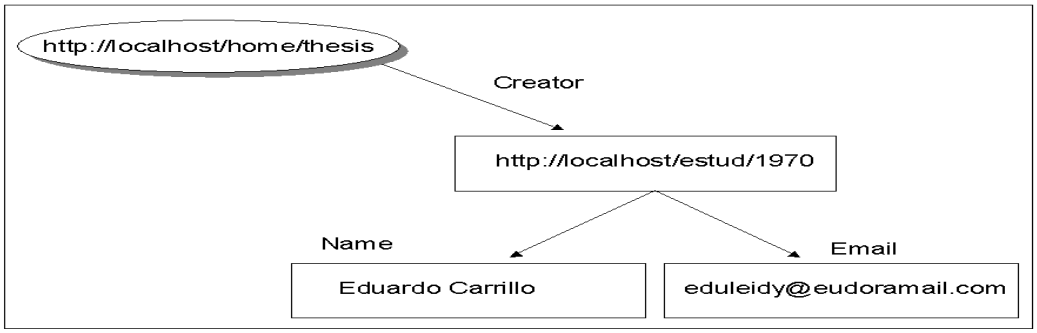


Figure 9. Expanded Example RDF Graph

RDF's conceptual model is a graph. RDF provides XML syntax for writing down and exchanging RDF graphs, called RDF/XML. Object Serialization is a process through which an object's state is transformed into some serial data format, such as XML or binary format, in order to be stored for some later use as information exchange in the network [Fox02]. Some of the models that make RDF serialization are RDF/XML and Notation 3(N3) a mechanism proposed by Tim Berners Lee, which is not a XML based format [Ber00]

RDF schemas are used to define the structure of the metadata that are used to describe WWW resources. The RDF Schema Specification consists of some basic classes and properties, and can be extended by others to fit possibly any given domain. Classes are sorted in a hierarchical manner, and the use of properties can be constrained to members of certain classes

RDF can use values represented according to XML schema data types, thus assisting the exchange of information between RDF and other XML applications. [W3C04b,W3C04c]

3.5.2.1.1. RDF Applications

This section describes some actual deployed RDF applications, showing how RDF supports various real-world requirements to represent and manipulate information about a wide variety of things. [W3C04b]

- The *Dublin Core* is a set of "elements" (properties) for describing documents (and hence, for recording metadata). The element set was originally developed at the March 1995 Metadata Workshop in Dublin, Ohio.

  Information using the Dublin Core elements may be represented in any suitable language (e.g., in HTML meta elements). However, RDF is an ideal representation for Dublin Core information

- *PRISM*: Publishing Requirements for Industry Standard Metadata is a metadata specification developed in the publishing industry.

  Magazine publishers and their vendors formed the PRISM Working Group to identify the industry's needs for metadata and define a specification to meet them. The PRISM specification uses XML, RDF, Dublin Core, and well as various ISO formats and vocabularies.

- Many situations involve the need to maintain information about structured groupings of resources and their associations that are, or may be, used as a unit.

  The XML Package (XPackage) specification provides a framework for defining such groupings, called packages. XPackage specifies a framework for

describing the resources included in such packages, the properties of those resources, their method of inclusion, and their relationships with each other.

The XPackage framework is based upon XML, RDF, and the XML Linking Language, and provides multiple RDF vocabularies:

- *RDF Site Summary RSS 1.0* is an RDF vocabulary that provides a way of describing information for timely, large-scale distribution and reuse. RSS 1.0 is also perhaps the most widely deployed RDF application on the Web. RSS 1.0 is extensible by design. By importing additional RDF vocabularies the RSS author can provide large amounts of metadata and handling instructions to the recipient of the file.

- *The Electric Power Research Institute (EPRI)* a non-profit energy research consortium, developed a Common Information Model (CIM).

  In addition, to further support the ability to electronically exchange CIM models, the power industry has developed CIM/XML, a language for expressing CIM models in XML. CIM/XML is an RDF application, using RDF and RDF Schema to organize its XML structures

- The objective of the *Gene Ontology (GO)* Consortium is to provide controlled vocabularies to describe specific aspects of gene products.

GO uses RDF/XML facilities to represent the relationships between terms in the XML versions of the ontologies, because of its flexibility in representing these graph structures, as well as its widespread tool support. At the same time, GO currently uses non-RDF nested XML structures within the term descriptions, so the language used is not pure RDF/XML.

- The W3C's Composite Capabilities/Preferences Profile (CC/PP) specification helps to address the problem of *Describing Device Capabilities and User Preferences* by defining a generic framework CC/PP.

CC/PP defines its structure (the hierarchy described above) using RDF Schema (see Device Independence section of this document for details)

Finally, RDF is also widely used as a representation format in many tools and projects as Amaya, Protege, Mozilla and SiLRI.

3.5.2.2. Semantic Web

Originally, the Web grew mainly around HTML, which provides a standard for structuring documents that browsers can translate to render these documents. On the one hand, HTML's simplicity helped spur the Web's fast growth; its simplicity seriously hampered more advanced Web applications in many domains and for many tasks.

Tim Berners-Lee first envisioned a Semantic Web that provides automated information access based on machine-processable semantics of data and heuristics that use these metadata. Some definitions of Semantic Web are:

- The conceptual structuring of the Web in an explicit machine-readable way. [Ber99]

- An extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation [Ber01]

- A Web where resources are machine understable and where both automated agents and humans can exchange and process information. [Sta02]

- The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.

With the advent of the Semantic Web, the current network of online resources is expanding mainly for a set of static documents designed mainly for human

consumption to a new Web of dynamic document, services, and devices, which software agents will be able to understand. [Ami02, Kot01]

Transforming the web into a "knowledge web" has put knowledge acquisitions and representation at the centre stage: Given the amount of available online information we already have achieved, this Knowledge (or Semantic) Web will be extremely useful and powerful. [Fen01, Noy01]

By associating meaning with content, the Semantic Web will facilitate search, interoperability, and the conception of complex applications. Specific instances of concepts defined in the ontologies -instance data - paired with ontologies constitute the basis of the Semantic Web. [Noy01, Thu01]

For example, consider an ontology describing roads, incidents and appropriate road-incidents combinations. Some of the classes describing this domain are Road and different type of Incidents. Some properties of the Incident class include initial kilometre, final kilometre and incident description.

The purpose of ontologies in the Semantic Web is to provide a kind of Semantic typing of the data distributed all over the Web to facilitate their interrogation by users through search or query engines, and more generally their use as input or output of Web services. [Cau02]

Typically, a Semantic Web Language for describing ontologies and instance data contains classes. Individuals in the domain are instances of these classes, and properties (slots) of each class describe various features and attributes of the concept. [Noy01]

In recent years, several markup languages have been developed with a view to creating languages that are adequate for realizing the Semantic Web. The construction of these languages is evolving according to a layered approach to language development.

Unfortunately, the relationships between adjacent layers in the semantic web model are not specified, especially that between RDFS and ontology languages, e.g., DAML+OIL. (See Figure 10)

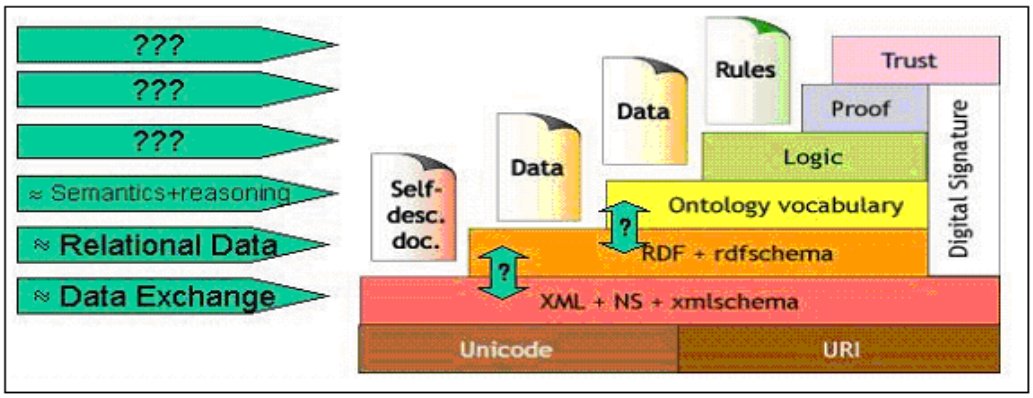


Figure 10. Web Semantic Layers [Pan03] [Ami02]

XML was the first language used to separate the markup of Web content from Web presentation, facilitating the representation of task-and domain-specific data on the Web. Unfortunately, XML lacks semantics. As such, computer programs cannot be guaranteed to determine the intended interpretation of XML tags. For example, a computer program would not be able to identify that <ROAD> data refers to the same information as <HIGHWAY> data, or that the <incident> specified at a Web service traffic service's site might be different from the <incident> at the police's site.

RDF, developed by the W3C for describing Web resources, allows the specification of the semantics of data in a standardised, interoperable manner. It also provides mechanisms to explicitly represent services, processes, and business models, while allowing recognition of no explicit information.

However, XML and RDF Schema combinations might solve this multiple-structure problem, but they would still lack expressive power [Wuw01]. For example, it is not possible to specify axioms, conditions and constraints. You can not define properties of properties, necessary and sufficient conditions for class membership, or equivalent and disjointness conditions for class membership, or equivalence and disjointness of classes.

The RDF data model does not provide mechanisms for defining the relationships between properties (attributes) and resources - this is the role of RDFS. RDFS offers primitives for defining knowledge models (for creating ontologies).

The RDF Schema takes a step further into richer representation formalism and introduces basic ontological modelling primitives into the Web. With RDFS, we can talk about classes, subclasses, sub properties, domain and range restrictions of properties and so forth in a Web-based context.

Initially RDF and RDFS had no formal model theory, nor any formal meaning at all. RDF Schema (RDFS) has a non-standard and non-fixed layer meta-modelling architecture, which makes some elements in the model have multiple roles and multiple-modelling primitives in the RDFS specification. As a result, this can be confusing and difficult to understand and, more importantly, the specification of its semantics requires a non-standard model theory.

This leads to semantic problems when trying to layer conventional first order languages, like DAML+OIL, on top of RDFS [Pan03]. The realization of the Semantic Web is underway with the development of new AI-inspired content markup languages, such as: [See Table 2]

Language	**Description**	**URL**
SHOE	Simple HTML Ontology Extensions	www.topicmaps.org
Topic Maps	ISO standard for describing knowledge	www.topicsmpas.org
XOL	XML-based ontology-exchange Language	http://www.ai.sri.com/~pkarp/xol/
DAML+OIL	DARPA Agent Markup Language +Ontology Inference Layer	www.daml.org
OWL	Web Ontology Language	http://www.w3.org/TR/2004/REC-owl-ref-20040210/

Table 2. Languages for the Semantic Web

We need a Semantic Web to express information in a precise, machine-interpretable form, so software agents processing the same set of data share an understanding of what the terms describing the data mean. [Ber99]

A more detailed discussion of the Semantic Web and its use in traffic systems is out of the scope of this thesis and it is also the main problem of another thesis that is being developed in the LISITT group of the Institute of Robotics.

3.6. DEVICE INDEPENDENCE

The device independence has as its main objective allowing the Web to be accessible in anyway, at anytime and through multiple access mechanisms. The foundation is based on the fact of being able to associate the Web content with the necessities, capacities and limitations of the used means. [But01, W3C03b,W3C04a]

The range of capabilities for input and output and the range of markup languages and supported networks greatly complicate the task of authoring web sites and applications that can be accessed by users, whatever device they choose to use.

Device independence encompasses the techniques required to make such support an affordable reality. In particular the device independence activity is focused on: [W3C02e]

- Methods by which the characteristics of the device are made available for use in the processing associated with device independence.

- Methods to assist authors in creating sites and applications that can support device independence in ways that allow it to be widely employed.

Generation of device independent Web content is being addressed in a number of several ways including:

- Multiple sources: under this scheme, the author creates different versions of the same Web information, each one with different characteristics according to the target access mechanism. In this way, many presentations for the same information must be developed, maintained and updated. When a request is received, the appropriate format for the information is selected based on the distribution context (such as device capabilities, user preferences and other restrictions).

- Stylesheets: the author creates a single set of information sources that are independent to any specific characteristic of the possible target devices. This information can be stored in different formats (ASCII, XML or other codifications).

  The author also creates groups of stylesheets dynamically for each access mechanism, for example using CSS (Cascade Style Sheets) or XSL (Extensible Stylesheet Language) . Under this scheme they must develop, maintain and update multiple variants of transformations and stylesheets [W3C00e]

  When a request is received, the respective variant for the device is generated through the combination of content and the appropriate template. Alternatively, the content is returned along with the appropriate stylesheet, and the user agent of the device is responsible for the presentation based on the received style sheet.

- Transformations: the author creates only a set of content in a markup language such as XML or XHTML. This content is transformed into another markup language according to the access mechanism. The transformation can be automatic, dynamic or static (using technologies like XSLT-Extensible Stylesheet Transformations), and can be carried out by the server or in an intermediate proxy. The range of transformations can vary from merely syntactic between markup languages, to semantic mapping of the available elements in a language. The content is transformed in the server or proxy, before being sent to the client. [Atz02]

- Object Oriented Programming: the author creates a set of components of re-usable objects and logical programming associated for its combination and execution. Depending on the type of device, the appropriate objects are combined and the logic is executed in real time, to generate the required output.

- Mixed Solutions: with complex contents the previous solutions are combined. Therefore, the author has to qualify and to implement multiple mechanisms to develop and to generate content.

  Depending on the type of content, a combination of adaptations can be necessary.

- Single Authoring: The author writes an intermediate abstract representation of the application. This intermediate representation describes the interaction of the user independently of the access mechanism.

- It also supports additional meta-information for a specific channel. This can be made by providing online annotations that affect the adaptation process, or overwriting the re-usable style sheets for this mechanism of access.

  At run time, the intermediate representation is adapted within pages of functional presentation, using stylesheets that only depend on the distribution context.

3.6.1. Requirements for device independent access to the Web

The requirements that must be fulfilled for access to the web in a device independent form can be divided into two main areas: The service provider point of view and the user point of view. The service provider providing the content wants it is as visible to as many users as possible, at the lowest cost. The user of the device is concerned with accessible content that their browser is able to unfold. [Boy03]

3.6.1.1. Service Provider's Perspective

From the service provider's perspective the objective is "write once, Render everywhere" (See Figure 12)

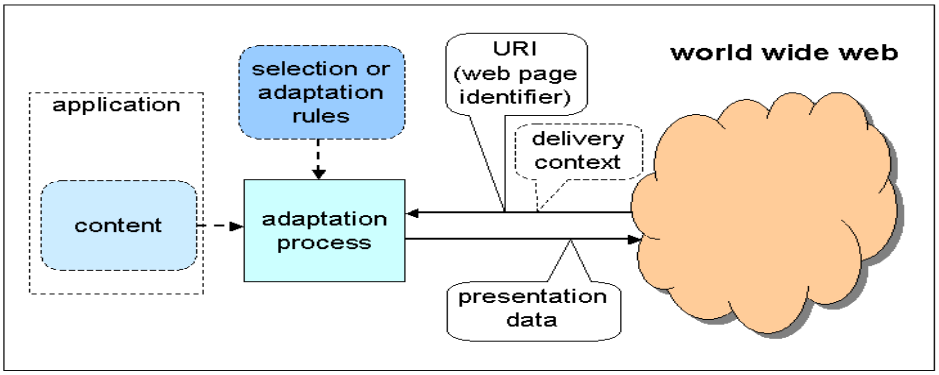


Figure 11. Service Provider's Device Independence Perspective [Boy03]

The service providers are the people or companies that usually create web sites; the web sites are created for different reasons. They can be created for economic reasons (for example, sale or announcement of goods or services), or for non-economic reasons (for example, to provide information).

When the reason is economic, the service providers wish to maximize utilities. This is obtained by integrating two objectives: to maximize income, and to diminish costs.

If potential clients can only buy products or see information about a company in a specific browser, any client who uses a different browser (for example, WAP telephone instead of PC browser with all the capacities) will not be able to see this information. By excluding a percentage of browsers, the companies restrict their potential market. Due to this, the service providers would ideally like their web sites to be more accessible from all the browsers. [W3C00f]

At the moment, if a service provider wants pages that are visible, for example on both a PC and a WAP telephone, he must write two different web pages. Although the content expresses the same information, two different sets of pages have to be written.

This increases the cost, compared with writing a set of pages that is viewed by any browser.

If a service provider could use a data source and some method or scheme to turn data "to the flight" to a form that all the browsers can see, it would only need to create sets of content, which would reduce its costs.

When the reason is not economic, the web site exists to provide information more than to make money. These sites will want to maximize their number of visitors. Their creator think that the information they are providing is useful and for that reason they whish that as many people as possible to see it.

In order to maximize the number of visitors to a site it is necessary to make it accessible from any kind of browser. If any browser is excluded, the users will not be able to view information.

3.6.1.2. User's Perspective

From the user's perspective, the web should be accessible by anyone, anywhere, at anytime, anyhow (See Figure 11)

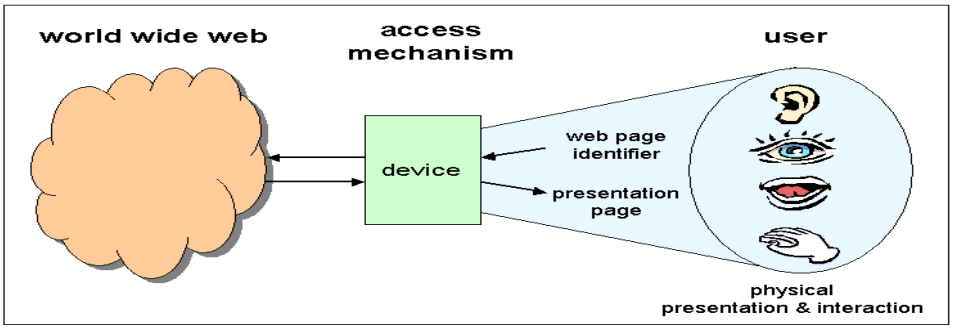


Figure 12. User's Device Independence Perspective [Boy03]

Users make many diverse demands of the web, and these can change depending on their navigational preferences or the selected browser.

People surf the web for many reasons. Some of them are educational other leisure related. Whatever may be the reason what the user wants is content, this content can be both varied in type and in many languages, using a diversity of protocols.

The users do not want to have to deal with technical details, but they want the same information to be available from any web browser. (For example, they want to be able to consult the availability of tickets for theatre from his WAP telephone, his TV adapter at home, or from a web browser at the office).

Although the information provider needs the creation of different web pages, the users wants to have access to and the same information, no matter the device or access procedure they use.

Even though the capacity of handling of content of the different navigators may be different, the users still want to be able to read the latest news headlines. Even though they can or cannot see the corresponding videos or listen to the voice reports, they still want to be able to read the news.

The users may have different preferences when they visualize web pages. This is for different reasons as the following:

- The user realizes that there is an inverse relation between the wealth of the content and the download speed to the device.
- The user may have personal preferences, such as not wishing to hear sounds since he is in an office with a PC that does not have audio output.
- A handicap which forces to the user to visualize the web through means (for example, requiring a voice enabled web browser)

If the user suffers from some form of handicap, he may have certain preferences that facilitate web browsing.

For example, a blind person wants the content read to him by the computer, and to be able to surf using voice commands, or a car driver may want to listen to his mail on his way to the office. So that a supplier provides information that can be visualized (in the broadest sense of the word) by anyone, the physical restrictions that some handicapped people have must be considered.

Also, for economic reasons, considering the cost of third generation wireless systems based on the amount of downloaded information, a user can choose to deactivate the graphical capabilities, and consequently diminish the costs of access to the textual information.

In conclusion, users expect to access the same content from any device with similar capabilities. Even, when device capabilities differ, user might still want access to an adapted version of the content.

Due to the diversification of wireless devices accessing the Internet, it is necessary to provide extensive information on the wireless device so that edge services can be carried out efficiently.

3.6.2. Delivery Context

A general solution for addressing the problem related with device heterogeneity, is that the client encodes its delivery context

Delivery context includes the device's capabilities, the user's preferences, the network characteristics, etc. - in such a way that a server can use the context to customize content for the device and user.

Delivery Context is “a set of attributes that characterizes the capabilities of the access mechanism and the preferences of the user " [W3C02d]

Some means to define the delivery context are the following: [W3C02c, W3C02d]

- CC/PP is a specific data format for expressing delivery context information (CC/`PP is described in section 3.4.3.3 of this document**)**

-Content Negotiation Working Group (CONNEG) format focused on content negotiation inside and outside of HTTP. (Content negotiation is the mechanism for selecting the appropriate representation when servicing a request). The feature set are mathematical relations, which define constraints on feature handling capabilities.

- Resource Capability Protocols (RESCAP): Protocol for the purpose of lightweight protocol (only a few bytes per transaction).

- Session Description Protocol (SDP): This is intended for describing multimedia sessions for the purpose of session announcement, session invitation, and other forms of multimedia session initiation.

- Pluggable Edge Services (OPES): This solution provides the means for a server to delegate the content adaptation to an intermediary in such a way that it does not break the end-to-end model of the Internet

- MPEG (Moving Picture Experts Group.) framework, which is intended to support transparent use of multimedia resources across a wide range of networks and devices.

- Transparent Content Negotiation (TCN) uses both HTTP server-driven and agent-driven negotiation mechanisms (as described above), together with a caching proxy that supports content negotiation

- SMIL (Synchronized Multimedia Integration Language) is an XML-based language that allows authors to write interactive multimedia presentations

- CSS Media Queries: A Media Query consists of a media type and one or more expressions to limit the scope of style sheets. Among the media features that can be used in media queries are "width", "height", and "colour". By using Media Queries, presentations can be tailored to a specific range of output devices without changing the content itself.

In solutions that provide device independence, it is typically the case that the delivery context influences the content adaptation processes. The content author will optionally provide additional information to affect this adaptation within particular contexts.

Finally, due to device differences, the adaptation might not produce an identical presentation, but device-independence principles suggest it should be sufficiently functional to let users interact with it successfully.

3.6.3. Adaptation Techniques

For optimisation reasons, i.e., in narrow-band channels such as the wireless optimisation, and for security reasons, it is more efficient to send only adapted information. It is possible to generate a style sheet that is adapted to the device dynamically.

Wherever the content adaptation takes place, it must be based on information about the delivery context.

Delivery context can include the delivery device's capabilities, the delivery network's characteristics, user-preferences, and other optional application-specific parameters such as user's preferred language or their location. CC/PP is a specific data format for expressing delivery context information (See section 3.4.2.3 CC/PP and UAprof)

There are three categories of adaptation techniques to approach the problem of device independence: [But02]

- Intermediate Level: It changes the presentation according to the capabilities of the client. For example: translation proxies. Theses are limited because they lack special information of the content. Their adaptation capability is limited. To translate HTML in WAP, HTML or HDML is possible only for some providers. On the other hand to translate pages that have hidden semantics, can produce unexpected results. The adaptation by an intermediary reduces the

load on the server but so much depends on knowledge of the target device and the metadata and adaptation characteristics developed by the author.

- Client Side: The CSS Media Types are names that identify different types of devices such as screens, handheld, TV, printers, Braille, etc. The CSS avoids many parts of web pages depending on the device, but this is not an effective use of means. Nevertheless, not all the devices support client side transformations.

- Server Side: This enables the author to have maximum control over the content and has the ability to change the amount of content, the style, navigation and layers. This technique is based on developing many versions of the sites of multiple versions and by technologies of development of scripts like Java Server Pages [Sun01], to make the transformations to send the appropriate content that can be visualized by the device.

From this group of techniques the one that offers the greatest control on the part of the developer of content is server side adaptation; this is reason why the proposals generated in this work are based on this choice.

3.6.3.1. User Agent

The headers sent with a HTTP request by a client form the base of most of the mechanisms of Delivery Context and of the transformation processes, and they are based mainly on the processing of user-agent. [W3C02g]

User agent strings are used by web clients to identify themselves when they send HTTP requests to web servers. [Zaw01, W3C00f, W3C00g]

The User-agent, or agent of a user, is a string that is sent by the client as part of a HTTP request it identifies the associated browser to the client request: Typically it includes the name of the manufacturer and the version number. Making an evaluation

of user-agent specific, means its content can be sent to a certain device, taking advantage of the string, which is unique for each.

Information about user agent includes the hardware platform, system software, applications, and user preferences. The user agent capabilities and preferences can be thought of as metadata, or properties and descriptions of the user agent's hardware and software.

Different definitions of user-agent can be found from the user-agent depending on the working context. In agreement with specification HTTP 1,1, user agent, it is defined as: "the client who initiates a request. They are often navigators, publishers, robots, or other tools of end user."

According to the Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies, W3C Working Draft of July 28th 2003, a user agent is defined as " A program, such as a browser, running on the device that acts on a user's behalf. Users may use different user agents at different times".

Additionally, by means of this mechanism the detection of some characteristics associated with the device is facilitated as supporting MIME (Multipurpose Internet Mail Extensions) types by a user agent, as they are the set of characters, the codification and the language preferred by the user [Bor92]. Nevertheless there are no standards on the way user-agents must be created, this makes it difficult to establish a unified handling that facilitates the processing of these identifiers.

3.6.3.2. XSLT (XML Stylesheet Language Transformation) [Atz02, Aki02, Mar01,W3C99f]

For content that is written in XML, it is possible to use the transformation mechanism using XSLT.

An XML processor (often called a parser, but called a processor in the XML recommendation) reads a source XML file and identifies the syntactic units (such as elements, attributes, and text content).

The XSLT processor read the XML and performs the actual XSL transformations. The XSLT processor takes a style sheet and applies it to the tree representation of a source XML document (produced by an XML parser) and generates a tree representation of an output document.

Conceptually, the XSL processor begins at the root node in the source tree and processes it finding the template in the style sheet that describes how that elements should be displayed. Each node is then processed in turn until there are no more nodes left to be processed. In fact, it is a little more complicated that this situation, because each template can specify which nodes to process - so some nodes might be processed more than once and some might be processed at all.

Because XSLT transformations operate on tree-like document data model described in the XPath specification [W3C99e], inputs and outputs to the XSLT processor will be represented as a tree structure. These trees often start and end life as documents in an XML notation. In other words, it might look like we are taking the document and changing the tags, but that is not how it works. XML tags (elements, actually) are not a way of representing underlying data structures.

The product of the XSLT processing is a "result tree". If the result tree is composed of XSL formatting objects, then it describes how to present the source document. There is nothing, however, that says that the result tree has to be composed of XSL formatting objects. It can be composed of any elements, and it does not have to be XML. When HTML is used in the result tree, XSL will transform an XML document into an HTML document.

Popular XML processor implementations produce either a stream of SAX events or a DOM object. In other words, the XSLT processor is dependent on the XML processor. It will take the stream (or DOM objects, but most XSLT processors use streams) and uses that information in a node tree (the style sheet tree) to create a new node tree (the result tree), possibly using several other node trees as input (the source tree or trees).

While checking source documents for validity can be very useful for diagnostic purposes, all of the hierarchical relationships of content are based on what is found inside the specific documents that is being input into the processing (the instance).

An XSLT style sheet is independent of any DTD or other explicit schema that defines the abstract model of the instance. In other words, XSLT can process well-formed XML that does not have an explicit data model (for example DTD or schema).

If there is a DTD or an XML Schema describing the document, however, certain information such as attribute types and defaulted values can be used to improve the processing. Without this information, the processor can still perform style sheet processing, however the absence of the information could not influence the desired results.

Because we do not depend on a specific DTD for the current document, we can design a single style sheet that can process different (but similar) documents.

The XSLT processing can be done in the browser but also in the server. It has turned out; it can be done more efficiently in the server, specifically for formats that are to be displayed in devices that are less capable in terms of display and processor (such as a mobile phone).

Contrary to the case of the browsers, where designers often feel that it is necessary to create a specific version of the document for each browser, the programmer can create an XSLT transformation sheet that will work on all processors by avoiding the differences.

3.6.3.3. CC/PP and Uaprof

For the adaptation of content and presentation to be effective, there has to be a way to describe both the content of a document and how to match it with characteristics of the device. Composite Capabilities/Preference Profile (CCPP) provides the equivalent of database fields and associated model for formalizing the device profiles. [W3C99a, W3C99b,W3C99c,W3C00a].

The Composite Capabilities/Preference Profile (CCPP) protocol is a particular application of RDF metadata and is based on a description of the capacities of the devices and the known user preferences like the user agent profile.

CC/PP is designed specifically to fit into the Semantic Web. It is based on RDF and is extensible with new components. (See Figure 13)

A profile is an instance of the vocabulary. Different devices and user agents may refer to the same schema and support the same vocabulary but communicate different profiles to the origin servers. Any vocabularies can be used in CC/PP, provided there is an RDF Schema for them, because the RDF processor uses this requirement.

This profile is sent by the client to the server to determine the most appropriate way to create, adapt or select the content presented to the client device.

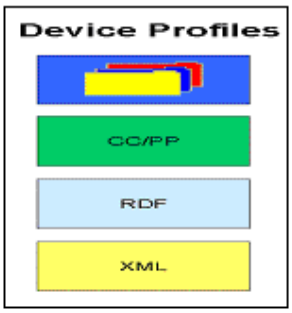


Figure 13. CCPP as RDF Application

The goal of CC/PP is to establish a vehicle for making this information available. Based on the Resource Description Framework (RDF), CC/PP provides a framework for the development of vocabularies by which devices can convey their capabilities when making a request of a server

CC/PP's goals include:

- Enhance content negotiation speed through a standardised format for user agent profiles.

- Minimise content negotiation transaction through the use of standardised formats and referencing URLs.
- Recognize and support the composition of preferences and profiles originating from multiple resources.
- Enable user control over user agent information.
- Support the presence of multiple network elements (proxies, servers, etc.) between the user agent and the origin server.

CC/PP defines a two-level component and attribute/value pairs hierarchy. A component can be used to capture part of a delivery context (for example, characteristics of the network, software supported by a device or the hardware characteristics of the device). [W3C03a,W3C00b, W3C00c,W3C00d]

A component can contain one or more attributes. For example a component that codifies the preferences of the user can contain an attribute to specify if is preferred audio output. Nevertheless, CC/PP by itself does not establish a vocabulary, but that is a generic language to construct such vocabularies.

The following example shows part of a CC/PP profile to specify user-preferences:

```
 <ccpp:component>
  <rdf:Description rdf:ID="UserPreferences">
   <rdf:type
rdf:resource="http://www.example.org/profiles/prefs/v1_0#UserPreferences"/>
   <ex:AudioOutput>Yes</ex:AudioOutput>
   <ex:Graphics>No</ex:Graphics>
   <ex:Languages>
    <rdf:Seq>
     <rdf:li>en-cockney</rdf:li>
     <rdf:li>en</rdf:li>
    </rdf:Seq>
   </ex:Languages>
  </rdf:Description>
```

```
</ccpp:component>
```

There is no limit on the number of vocabularies that can be created or used. In fact, RDF and XML namespaces enable independent creation of interoperable vocabularies. Schema interoperability, enabled by RDF and the use of XML namespaces, enables the selection of any number of attributes within the context of a user agent profile, because it becomes possible to select any element that is present in a schema using the XML Namespaces mechanism.

Vocabularies that conform to CC/PP have to be created as an RDF Schema, made available at a URI, and be usable in a component in the CC/PP framework. Vocabularies can be included in CC/PP using XML namespaces. If a vocabulary is not written in RDF Schema, and does not conform to the CC/PP component structure, it will not be possible to use it in CC/PP.

On the other hand the Open Mobile Alliance (wapforum) has defined the user agent profile (UAProf) - a framework based on CC/PP that includes a vocabulary to describe capabilities of the device, capabilities of the user-agent, network characteristics, etc., as well as a protocol to transport a profile [OMA02].

UAProf defines six components including: HardwarePlatform, SoftwarePlatform, NetworkCharacteristics, WapCharacteristics, PushCharacteristics and BrowserUA. Also it defines several attributes for each one of its components although the attributes of a component are not fixed - they can be replaced or eliminated. The following example shows part of the Ericsson T68's UAProf profile

```
<?xml version="1.0" ?>
 <RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prf="http://www.wapforum.org/UAPROF/ccppschema-20000405#">
 <rdf:Description ID="Profile">
 <prf:component>
 <rdf:Description ID="HardwarePlatform">
  <prf:ScreenSize>101x80</prf:ScreenSize>
```

```
 <prf:Model>T68R1</prf:Model>
<prf:InputCharSet>
<rdf:Bag>
 <rdf:li>ISO-8859-1</rdf:li>
 <rdf:li>US-ASCII</rdf:li>
 <rdf:li>UTF-8</rdf:li>
 <rdf:li>ISO-10646-UCS-2</rdf:li>
 </rdf:Bag>
 </prf:InputCharSet>
 <prf:ScreenSizeChar>15x6</prf:ScreenSizeChar>
 <prf:BitsPerPixel>8</prf:BitsPerPixel>
 <prf:ColorCapable>Yes</prf:ColorCapable>
 </prf:WtaiLibraries>
...
 </rdf:Description>
...
</prf:component>
...
 </rdf:Description>
 </RDF>
```

Finally, some CC/PP implementations include:

- X-Smiles: X-Cries CC/PP Servlet.
- Jisgaw: W3C's CC/PP capable Server with HTTP extension.
- DELI: Open source delivery context Java library supporting CC/PP and UAprof.
- DICE: Device Independent Content Engine, an opensource CC/PP and UAProf compliant web server extension. [W3C02f]
- WebDAV : Web-based Distributed Authoring and Versioning. It is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers.

3.6.4. Available Solutions for Ubiquitous Content Generation

3.6.4.1. Opensource Library DELI (Delivery Context Library) and capability classes

DELI is a library developed by HP that provides an API (Application Programming Interface) for Java to process CCPP and UAProf information contained in a HTTP request. [HP01]

Acting as a middle-agent, DELI performs profile resolution, reassembling packets and then querying profiles in the profile repository. It also allows the manipulation of CCPP vocabularies that have not yet been created.

When DELI interprets a request of a device that does not come accompanied with the information from CC/PP or UAProf, it uses the agent-user chain to identify the device in a database of previous devices and then it provides the profile associated with the device

Since it supports legacy techniques, such as the use of conventional HTTP header information, as well as CC/PP and UAProf, it enables server-side applications to be developed now that can support both the old and new approaches. In this way it smoothes the path towards adoption of the new standards.

With the development of DELI, HP proposed a denominated mechanism Capability Classes, by which means the designer defines a series of device classes and the restrictions to handle the profiles.

This mechanism allows the creation of groups according to their similarity. These groups can be registered in a file that contains the restrictions and logic expressions that allow its creation.

A number of capability classes are defined where each class is associated with a set of constraints. When a server receives a profile, it evaluates each set of constraints to determine if the target device belongs to one or more of the capability classes. Once it

has determined which capability classes are supported by the device, this information is passed to the stylesheet to guide transformation. The same mechanism may also be used for other types of content specialisation such as selecting stylesheets, performing content negotiation or content generation.

Using capability classes instead of user agent chains reduces the number of mappings required due to the form in which capability classes can be used to generalize through devices. Alternatively capability classes can be used to transcode resources e.g. an image could be transcoded to change the target format and resolution.

The capability class examples use the data types available in UAProf. The CC/PP (or UAProf) processor uses the capability class description file as follows: it parses the file and constructs a postfix description of each set of constraints. It stores this postfix description in a vector for evaluation later.

The processor evaluates the postfix description of a set of constraints by retrieving each operator or operand in turn from the vector; evaluating it and then writing the result back to a results stack. The processor then determines if CcppAccept contains the value of the property and writes the result to the results stack. Then it pops the previous three values from the results stack, applies the AND operand and writes the result back to the result stack. This indicates if the device is a member of the capability class. The processor repeats this process for the postfix description of each capability class and returns a vector containing the names of any capability classes where the device meets the associated constraints.

Capability classes make it possible to classify the capabilities of different devices and use these classifications with XSLT or with other content specialisation mechanisms such as configuration files that describe how resources are selected or generated via transformation and transcoding. They have a number of advantages over the previous device class methods for classifying devices.

A sample implementation of capability classes for evaluating this technique is currently available in the DELI library available within the Cocoon [Apa02] framework

Capability classes are the main conceptual proposal that has been found in relation to the grouping of devices that allow device independency. Nevertheless, and as far as this study goes, it cannot offer a proposal of the form the designer can create these groups.

3.6.4.2. Commercial Solutions

3.6.4.2.1 Mobile Internet Toolkit

The Mobile Internet Toolkit (or Mobile Controls) proposal has been developed by Microsoft within the general framework known as Dot Net [Mic01,Mic02]. With this approach the development of interfaces is made by means of graphical tools using drag and drop techniques, without worrying about the markup language that is used in the design. The server is in charge of making the appropriate transformations and generating the content for a determined request, from the interaction with a previously defined (and updateable) list of devices that is in the server. Nevertheless, the disadvantages of this approach include that it does not use proposals for content and presentation separation (like XSL and CSS), and it is only possible to use it in Microsoft based servers.

3.6.4.2.2. Oracle 9i Application Server Wireless

Another one of the commercial solutions has been developed from the database world by Oracle with its proposal Oracle 9i and Oracle 10g [Wad01], that includes an proxy architecture, support for handling diverse formats and intermediate XML transformations that are available for multiple devices through the solution known as "Oracle 9i Application Server Wireless" and previously with its proposal Portal-to-go.

3.6.4.2.3 IBM Websphere Transcoding Publisher

On the other hand IBM has developed the proposal InfoPyramide, which is included within its product IBM Websphere Transcoding Publisher [IBM01,IBM02], that provides a general framework with development that also includes support of diverse

formats, with a base in a transformer of content, based on images using an proxy architecture.

A common disadvantage of these approaches is dependency on the manufacturer. It would be advisabale to consider the possibility of future platform changes.

Along with the development of DELI, HP proposed a mechanism named Capability Classes, by means of which the designer defines device groups or classes and its restrictions to handle the profiles. This mechanism permits groups of devices to be made in agreement with classes of similar characteristics, which can be registered in a file that contains the restrictions and logic expressions that allow the creation of groups.

3.6.5. Evolution of the Device Independence and CC/PP in W3C [W3C02b,W3C03b]

W3C organizes the work necessary for the development or evolution of a Web technology into activities. To manage related activities, the W3C Team groups them into four "domains": Architecture, Interaction, Technology and Society, and the Web Accessibility Initiative.

- The Architecture's goal is enhancing the infrastructure of the Web and increasing its automation.

- The Interaction Domain is responsible for developing technologies that shape the Web's user interface.

- The Technology and Society Domain's goal is to augment existing Web infrastructure with building blocks that assist in addressing critical public policy issues affecting the Web.

- The Web Accessibility Initiative Activity includes work on technology, guidelines, and tools to increase accessibility of the Web for people with disabilities

The W3C publishes two types of technical reports: ones which make recommendations that are specifications, guides, etc., produced by the workgroups; and notes that constitute a public registry of an idea, commentary or document, and are published by the discretion of the director

In order to turn a technical report into a recommendation, the following process is required (See Figure 14):

- A Working Draft is a document that W3C has published for review by the community, including W3C members, the public, and other technical organizations. These are draft documents and may be updated or replaced by other documents at any time. It is inappropriate to use W3C working drafts as reference material or to cite them as other than "work in progress".

- A Candidate Recommendation is a document that W3C believes has been widely reviewed and satisfies the Working Group's technical requirements. The W3C publishes a Candidate Recommendation to gather implementation experience.

- A Proposed Recommendation is a mature technical report that, after wide review for technical soundness and implement ability, W3C sends to the W3C Advisory Committee for final endorsement.

- A W3C Recommendation is a specification or set of guidelines that, after extensive consensus building, has received the endorsement of W3C Members

and the director. W3C recommends the wide deployment of its Recommendations. Note: W3C Recommendations are similar to the standards published by other organizations.

Figure 14. Process of a W3C technical report [W3C00h]

A Working Group Note is published by a chartered Working Group to indicate that work has ended on a particular topic. A Working Group may publish a Working Group Note with or without its prior publication as a Working Draft.

The evolution of the Composite Capability/Preference Profile technical report until it acknowledgement as a W3C Recommendation is outlined in Figure 15. Figure 16 presents the process followed by the Device Independence Principles.

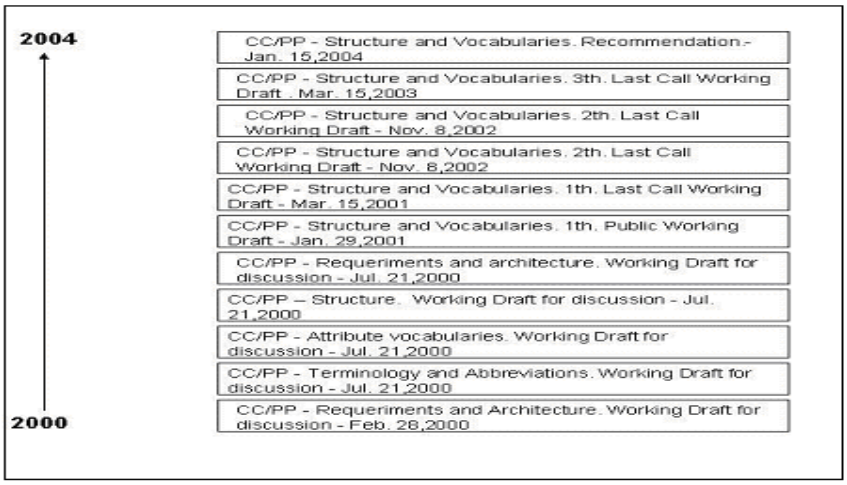


Figure 15. Evolution of CC/PP technical report

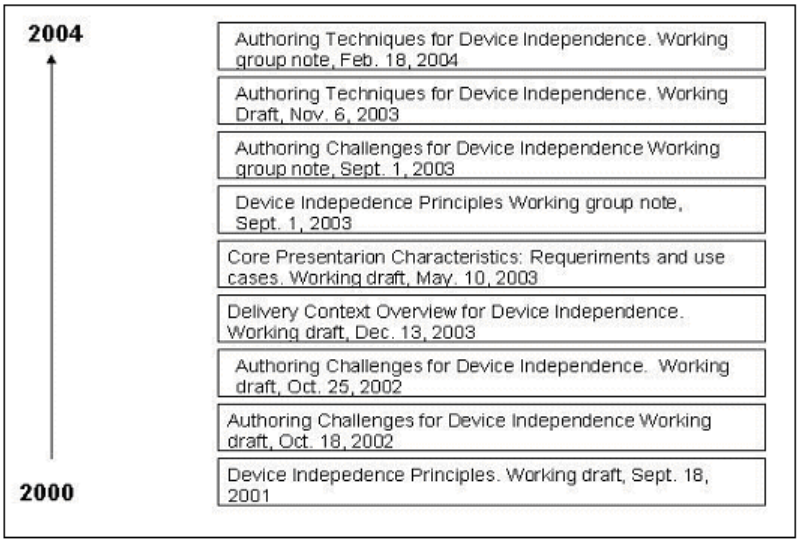


Figure 16. Evolution Device Independence technical report

3.7. CONCLUSIONS

Starting from the analysis of current proposals for content generation independently from devices, generated by consortiums such as W3C, it can be seen how the principles that give way to these proposals stem from different areas of computational science.

For example, the description of device profiles based on vocabularies such as UAProf, and the generalized frame for describing CC/PP vocabularies that constitute an RDF

application, but at the same time have their origins in the creation of ontologies and semantic networks which have been generated by communities such as that of knowledge representation.

In relation to the techniques of software engineering, they have rapidly evolved, allowing inclusion of specific component treatment of Web applications, such as hypertext, session management, as well as different levels of abstraction that include context concepts and diverse dimensions.

From these proposals only a few count on elements related to heterogeneous applications such as those based on multi-device interaction, and in those that exist there is no clarity within the existing link between the logical and the physical modelling.

It has been proved how the CC/PP protocol is constituted a clear RDF application producing a useful approach for the processing of device profiles. In relation to the possibility of using XML schemes for the device description, they only provide the profile description with syntactic elements, they are not prepared for a semantic treatment as it occurs with descriptions based on RDF.

Following that W3C proposals related to the problem of Web content access from multiple devices are described. These proposals are based on the models for metadata description and the search for a semantic Web that goes beyond simple markup. Finally the device independence proposal generated by W3C was described.

After introducing the main concepts related to the development of multiple devices based on device independence principles, several existing commercial solutions for application development that are based on multidevice content generation are described. The commercial solutions for device independence area characterized by presenting solutions that are completely dependent on the manufacturer.

The Capability Classes were described, which poses the need of device groups for content generation and includes the possibility of defining these groups, which can be treated making use of the free distribution of the DELI library.

4. TRANSPORT TELEMATICS TECHNOLOGIES

4.1. SUMMARY

In this chapter, concepts and principles related to ITS (Intelligent Transport Systems) are presented. This chapter serves to complement to the conceptual framework that has been generated for the prototype and application development that are described later in chapters 8 and 9, since all of them are Traffic Information Systems that allow the ubiquitous access to traffic incidence information.

Concepts such as traffic data, traffic information and situation are described, as well as the different levels in the information chains related to traffic. Following, the existing functional areas existing in the traffic information systems are listed, as well as the different technologies involved in these activities. Finally, an outline of the benefits of using information technologies in traffic information systems is presented.

4.2. INTRODUCTION

The first telematics systems, which appeared in the late 1960s, were computerised signal control systems designed to optimise urban traffic flows. Over the years, a growing number of increasingly sophisticated products and systems have become available.

For this reason the traffic information has to be managed more efficiently on existing roads, and traveller need to be encouraged to use new sources of information. It is important that all the systems included in a modern system for traffic management and control system are fully integrated.

Intelligent Transport System (ITS), also referred as transport telematics, include a wide range of tools and services derived from information and communications technologies, and management strategies, in an integrated manner, to improve the safety, capacity and efficiency of the transportation system. [Pou01]

4.3. DEFINITIONS

Throughout the traffic community, the terms “data”, ”information”, “service provider”, “content” and so on are freely used but are often mixed and are used without real understanding of what they mean. The following definitions are taken from the DATEX Data Dictionary version 3.1. [Eur01]

• Traffic data: Traffic data means status data. A typical example is traffic flow. Also level of services can be considered as traffic data.

• Traffic information: Traffic information means event information. It is defined as disturbances to a ‘normal’ status. Typical examples include accidents and road works.

• Traffic situation: From traffic data and traffic information a traffic situation can be described (e.g.: accident during road works; all lanes black, 2 km congestion). A traffic situation is described by traffic information elements.

It is possible to consider three levels within the information chain:

• Content provider: An organisation that makes available to any other organisations traffic data or traffic information. They may have an original focus on traffic management. An example could be the motorway police.

• Publisher: An organisation that brings together traffic data and/or information from one or several content providers. This is done in order to create a clear view of the traffic situation. They may undertake traffic management. They provide the information to end user service provider.

• End user service provider: An organisation that passes on traffic information to end-users especially the public. These are increasingly becoming private sector services offering a broad range of service delivery channels (Internet, SMS, GSM, GPRS, TV, Radio, RDS-TMC) as well as established public sector services.

Organisations do not often fall into one single category. All broadcasters are end-users OR service providers. They broadcast traffic information but most of them also gather traffic data and information from different sources in order to have a clear view on the traffic situation.

4.4. TRAFFIC MANAGEMENT

A traffic management centre takes all measures to assure the safety on the road and to optimise traffic flow. Often the term TCC (Traffic Control Centre) is used as synonym. The TCC gathers traffic data and information and takes measures by e.g. setting Variable message signs (VMS) with variable speed limits or rerouting advice.

A TCC is in most cases linked to a TIC (Traffic Information Centre). In some countries several TCC's are linked to one TIC. Often the TCC and the TIC are integrated into one centre. The TIC is responsible for all exchange of traffic data and information in both directions. Therefore the information exchange for cross border management happens between the TICs of the neighbouring region.

The TIC brings together traffic data and information from different sources in order to have a clear view on the traffic situation for traffic management purposes. By doing so, they add value to content that is interesting and potentially valuable for end user service providers. In this case we consider the traffic centre as service provider.

In some cases, the TIC and TCC will be totally integrated and will also be responsible for some data collection. Hence there are a wide variety of roles across but still the main aim is that information that can be used to support end user services. [Com01]

4.5. FUNCTIONAL TRANSPORT AREAS

ITSs use in the transport sector has different appearances and concerns different modes. [Nij95, Pou01]

The first phase, one may distinguish IS the provision and use of information by means of telecommunications to trip-makers in order to increase the efficiency and reliability

of transport operations. The same holds for automatic debiting systems for parking or road pricing. In this case, telematics does not necessarily affect transport behaviour (in terms of route choice, trip scheduling, departure time), but it serves to increase the performance of transportation.

Next, we may mention various telematics technologies, which have an immediate day-by-day consequence for transportation behaviour. Examples are route guidance, variable message signs or radio data information. Such uses of telematics influences spatial behaviour of trip-makers, not only for car users and truck drivers but also for users of public transport.

And finally we may distinguish telematics applications, which have a structuring impact on mobility behaviour. This may be found in tele-working, tele-commuting, etc., where trip making (e.g., home-to-work) is influenced in terms of changes in commuting. It is also possible to split up the traffic information services in different classes according to the area:

- Regional service: the service only concerns a specific region.
- National service: the service concerns the whole country.
- Cross border service: the service concerns the own region and traffic situations in the neighbouring regions, but only traffic situations on a relative small distance from the border (say up to 50 km)
- International service: the service concerns the traffic situation in several regions not all belonging to the same country or even wider.
- Individualised service: the service responds to individual needs of the end-user.

An extended or amplified classification of the principal areas that form part of the transport telematic technologies, developed from the classification proposed by [Nij95], covers the following areas:

- Travel and Traffic Information
- Public Transport Management

- Freight and Fleet Management
- Road and Traffic Management
- Demand Management
- Parking Management
- Diver Assistance and Cooperative Driving
- Electronic Payment Systems
- Emergency Management Services
- Vehicle Safety and Control Systems
- Information Warehousing Services

However, many of the functions can be overlapped and belong to more than one area.

4.5.1 Travel and Traffic Information

The area of provision of travel and traffic information is closely related to the implementation of most other telematics functions: demand management, parking management, traffic management and public transport and freight operations.

The functional area of travel information has applications for all kinds of travellers; it includes the provision of information before and during the trip to facilitate travel planning and travel bookings, as well as information to drivers about services whilst driving.

The functional area of providing traffic information for vehicle drivers includes the provision of dynamic traffic and traffic-related information to drivers during the trip via either collective means or individual in-vehicle terminals.
The classical means to disseminate travel information are broadcasting by radio and TV (teletext). More advanced private terminals comprise personal portable units, in-vehicle units, and home and workplace units.
Travel information is supplied from multiple and diverse sources, e.g., public transport operators, rail operators and tourist organizations. These private and public systems have been available for some time and are being expanded to provide multimodal travel information.

According to the communication links between the vehicle and its surroundings, information systems to the driver are classified into five different categories:

Autonomous navigation systems. In these systems no external communication link exists. They are self-contained systems for providing drivers navigation assistance.

*Area broadcast systems***.** These systems have one-way link from control centre to vehicle. They enable traffic messages to be broadcasted over a wide area, typically to about a hundred kilometres in diameter.
One of the more developed broadcasts is the Radio Data System (RDS). This system enables digitally encoded messages to be superimposed on normal CHF-FM radio programme broadcasts. Depending on the in-vehicle software, RDS systems can provide all kinds of service information. The use of this technique for transmission of traffic information is known as RDS-TMC (Radio Data System – Traffic Message Control).

Local roadside transmitter systems represent a logical extension of the area broadcast systems to provide higher density of transmitters. Communications technologies used for such systems are mainly microwave and infrared techniques. They basically have three main uses: hazard warning-only transmission, location beacons and combined solutions to guide the vehicle.

Mobile radio systems with two-way communication links between a control centre and the vehicle. There are several mobile radio systems available. However, these systems are usually limited in terms of the number of vehicles they can serve.

The main difference with the previous technologies is the extra communication link from the driver to the control centre. This link provides new possibilities, ranging from emergency calls to interactive information. A very important implication of both cellular radio and satellite navigation techniques is that the driver can report back his position and status from time to time.

Local roadside transreceiver systems with two-way communication between roadside units and special in-vehicle units. These systems can provide in the first place autonomous navigation, using in-vehicle simple directional data. In a second place, they can act as complex beacons. Third, these systems have the capability of communicating their trip times to the infrastructure.

4.5.2. Public Transport Management

This kind of system is also referred as APTS (Advanced Public Transport Systems). They are applications to improve the efficiency and the quality of the service to the users of the public transport of information systems, automatic payment systems and systems of location.

In this field it can be distinguished five key technologies (some of them partially overlap the advanced transport information systems):

*Automatic Vehicle Monitoring (AVM) and Automatic Vehicle Location (AVL***)** *systems*. This kind of system includes hardware and software in vehicles and operating centres, and the communication links between vehicle, roadside and control centre. AVM systems cover a range of functions such us vehicle diagnostics and passenger counting, as well as fleet monitoring and control.

Interactive passenger information terminals. These terminals can be used by travellers or by the staff of enquiry offices and can be located at home, in the office, public places and enquiry offices. The information provided is generally static, i.e. based on current time tables, and can be mainly used for trip planning.

Real time passenger information systems are mainly used to provide real arrival and leaving times. They are used adjunct to AVM and AVL systems

Fare collection systems includes smart cards that can be used for integrated payment systems in which the same card is used for public transport and other purposes, e.g. car parking, telephones, etc.

Public transport priority systems include public transport priority lanes, gating systems and priority treatment for buses and trams at traffic signals.

4.5.3. Freight and Fleet Management

Some of the activities included in Freight and Fleet Management are based on effective planning and monitoring of the fleet. Furthermore, management systems exist which deal with dispatch, storage and customs documentation and delivery information. Other systems deal with post-shipment analysis of performance and invoice/payment arrangements. The last area of fleet operation related is to monitoring, recording and analysing vehicle and cargo status.

The following four main technologies can be distinguished:

Electronic Data Interchange (EDI*)* systems comprise the electronic exchange of transport documents, orders, etc. in a standardised form between road transporters, clients and receivers.

Automatic Tracking and tracing has the aim of making digitised data available to a computer without human processing. Automatic identification is a component of tracking technologies, which are an important part of many advanced communications, and information technologies being used to improve the efficiency of freight transportation.

Automatic Vehicle Location (AVL) and two-way communication systems make it possible for logistic operators to monitor movements of vehicles. AVL systems and satellite communications systems envisage software equipment both in vehicles and in the fleet operator centre for tracing proposes.

Navigation and route guidance systems are telematics applications that assist drivers on their trip with their navigation task, by providing data on current location and traffic conditions.

4.5.4. Road and Traffic Management

The management of roads and traffic is a task of national and regional responsible government infrastructure authorities and operators. It involved a wide series of relevant application as the following:

Monitoring Systems comprise inductive loop detectors and Close Circuit Television (CCTV) systems based on video sensors, which are being used for real time data collection on traffic counts, congestion and incident detection on roads.

Variable Message Signs are used to provide car drivers with various kinds of dynamic information, like information on road works, incidents, lane restrictions, weather, environmental and traffic conditions.

Ramp metering is the merging of a traffic flow onto a congested motorway by means of continuous one-by-one access controlled by a traffic light.

Emergency telephones serve to help drivers in cases of incidents or technical problems.

Priority Lanes are systems that enable specific road user groups to access part of the strategic network with some of priority of particular interest to policy makers. Possibilities include the priority treatment to buses at traffic lights or bus lanes on congested (motor) ways in order to simulate public transport.

4.5.5. Demand Management

This is matching of traffic demand with the available infrastructure. Telematics can provide means of managing traffic demand by various functions that can be divided in two main areas: demand restraints and supply control. Domain restraints includes: area access restriction, route diversions, road pricing, and area parking strategy. Supply control include car pooling and modal interchange.

4.5.6. Parking Management

Parking Management is an element from the fields of demand management, traffic control and travel information. Telematics functions in this field are related to reservation, debiting of parking charges and enforcement/penalty systems for both on and off street parking spaces.

4.5.7. Diver Assistance and Cooperative Driving

The functional area of diver assistance and cooperative driving embraces the monitoring of drivers, vehicles and surroundings, and provides direct or indirect assistance for driving, either by warning stimuli to the driver or by partial or full automatic control of the vehicle. The technologies included in this area are the following:

Automotive radar systems are the main option for the continuous monitoring of the distance and speed between vehicles on the road. Such radar systems consist of a transceiver fixed on the vehicle that continuously transmits signals that are processed and combined with information about the host vehicle. After that, algorithms process the data and determine if hazardous situation exist.

Driver status monitoring systems are designed to monitor the driver status. Driver parameters can be derived from physiological measurements by means of electro-encephalograms, electro-oculograms and video camera views of the driver's face.

History and instruction module systems aim to improve driver's behaviour by providing the driver with information about how well manoeuvres have been performed, in a way which takes into account the experience and past performance of the driver.

Dialogue management systems comprise enhanced man-machine interfaces, which manage and supervise all communication between the driver and various telematics systems

4.5.8. Electronic Payment Systems

The technologies included in this area can be classified in the following way:

Electronic Fare Collection (EFC) systems integrate electronic communication, data processing, data storage and microcomputer technologies into the process of fare collection, subsequent record keeping and funds transfer. New innovations include magnetic fare cards and smart cards containing a microprocessor.

Electronic Toll Collection (ETC) allows for electronic payment of highway tolls. ETC systems take advantage of vehicle-to-roadside communication technologies [Automatic Vehicle Identification (AVI) and sometimes Automatic Vehicle Classification (AVC)] to perform an electronic monetary transaction between the vehicle passing through the toll station and the toll agency.

Electronic Parking Payment. The major inconveniences associated with parking vary depending on the type of facility being accessed. Parking customers lose valuable time waiting for a space, to get a ticket or pay a cashier. If these delays are excessive, they can also cause problems for the travelling public if there is a back up into traffic thoroughfares.

Meter patrons risk the situation where they do not have the appropriate amount of change or no change at all. Electronic payment can eliminate the need to stop to get a ticket or to pay. Parking meters that take smart cards eliminate the need for having coins in hand.

4.5.9. Emergency Management Services

This techniques assist governments in making initial decisions upon arriving at the scene of a dangerous goods incident. They are primarily designed for use at a dangerous goods incident occurring on a highway or railroad.

The main goal is to reduce the number of deaths and injuries caused by collisions, as well as damage to property, the environment, and health resulting from the use of motor vehicles.

Some activities included are improving emergency vehicle response time by fleet tracking, route guidance, signal pre-emption, hazardous material planning and incident response, disaster response/management and emergency vehicle management.

4.5.10. Vehicle Safety and Control Systems

This technology is related with the car's evolution from a mechanical device to an increasingly computerized one, in which electronic impulses replace or augment moving parts. That means microprocessors can take control of the most basic driving functions, like steering and braking.

At the same time, there is a parallel evolution in sensory technology. Most advanced safety systems are equipped with sensors that look inside the car, tracking tire rotation, brake pressure and how rapidly a driver is turning the steering wheel.

These systems include in-vehicle technologies such as on-board computers, collision avoidance sensor technologies, etc. But next-generation sensors, including radar and hidden cameras, are looking outside the car, giving it the ability to open its eyes, so to *speak*, to its surroundings.

4.5.11. Information Warehousing Services

This function includes all the data collection and management capabilities provided by the ITS, and adds the functionality and interface definitions that allow collection of data from multiple agencies and data sources.

It also provides the historical data archive repositories and controls the archiving functionality for ITS. The Data Warehouse Distribution function integrates the planning, safety and processes data products.

4.6. BENEFITS OF ITS

Whether offering "real-time" information about current traffic conditions or on-line information for journey planning, ITS tools enable authorities, operators and individual travellers to make better-informed, more intelligent decisions.

With ITS, travellers can benefit from increased safety, better information, greater comfort and reduced journey times. ITS can help network and fleet operator offer a better service. ITS can help public authorities to implement policies and measures to ensure a sustainable transport system for the future.

According to and study done on behalf of the European Commission in collaboration with ERTICO (Intelligent Transport Systems Europe) the great challenge of intelligent mobility are:

On the one hand, ITS allows the construction of transport systems that can improve the mobility, safety, and the quality of life of citizens.

Through effective integration with the transportation system, ITS provide a broad spectrum of technology-based tools that aid the development of solutions to save lives, money, time and the environment.

ITS allow road authorities to better manage and maintain the transportation system and enforce regulations by providing real-time and reliable information. ITS technologies can build on each other, providing higher order benefits through system integration and assistance to multiple users.

As technologies change and evolve, additional benefits will continue to be identified. However, some examples are:

- *Improve Safety*: Pre-trip and en-route road and weather information systems can advise motorists of traffic, road, environmental and emergency conditions. Real-time information can assist route planning, ease frustration and reduce road rage. Congestion reduction measures can

reduce travel time and collisions. Surveillance systems and wireless emergency location systems can automatically detect incidents and trigger the emergency notification and dispatch process. The end result of these applications is the reduction in loss of life, injuries and costs, which benefits society as a whole.

- *Increase Productivity*: Automatic vehicle identification, screening of safety records and vehicle weights, can provide a seamless commercial vehicle system, increase the efficiency of the inspections services, and provide opportunities for commercial vehicles.

  Real-time integrated transportation system data collection can improve the efficiency of the data collection process and facilitate traffic forecasting and planning. ITS can provide information about transportation trends and the performance of the transportation system, which can lead to better management and operations, more efficient allocation of resources, and improved system performance.

- *Reduce Costs*: The integration of ITS into the existing transportation system is becoming simpler, cheaper and more cost effective. With increasing congestion and the rising costs of providing additional infrastructure, ITS offer innovative alternatives to capital improvements. Once integrated, ITS can increase the efficient movement of vehicles, thereby increasing roadway capacity and reducing or delaying the need for additional infrastructure.

- *Help the Environment*: Incident management systems can improve emergency clearance times and reduce vehicle delay. Automatic vehicle identification at inspection stations and border crossings can reduce the number of commercial vehicle stops and starts. Real-time adaptive traffic signal controls can reduce travel times and vehicle idling. The effect of these applications is a reduction in fuel consumption and gas emissions.

4.7. CONCLUSIONS

Today, the situation in end user services for traffic information is still rather focused on national markets and national targets, as the main end user service providers are heavily language dependant (broadcasters) or developed from national organisations (automobile clubs etc).

As already indicated before, for traffic management the situation seems to be much more clear and stable than for Travel Information Services. There are information exchange standards, agreed implementation profiles and network architecture.

For the ITS domain, there seems to be a completely different starting point. For end user Service Providers, it is difficult to work out a network architecture and it is not acceptable to start with a variety of different bilateral negotiation processes whose outcome, resource and time requirements cannot be pre-determined.

Processing traffic incidence information by means of ITS corresponds to different areas of telematics applied to transportation. This type of information can be of great utility for the processing of route and trip planning, which finally generate services to society.

Within these services, the possibility of counting on computational and communication tools that allow a better handling of time by the citizens is included, as well as a better use of route distribution by avoiding problematic routes, better planning on the part of merchandise distribution companies, and accident prevention.

On the other hand, access at any given moment, from any place and with any kind of device that is characteristic of ubiquitous applications constitute an added value that can be used by society, expanding the benefits that access to the information through Internet by conventional mechanisms presents.

Different ITS areas included in this chapter describe the benefits with real time access to information of traffic incidences. This serves as theoretical base for the prototypes and applications that have been implemented in this dissertation.

The work of this thesis therefore contribute to the international traffic information services by supporting the establishment of these services and overcoming accessibility barriers as language or device used to get the information.

SECTION II: PROPOSALS AND CONTRIBUTIONS

FRAMEWORK FOR UBIQUITOUS WEB APPLICATIONS DEVELOPMENT

This proposal is based on the capacity to model personalization elements in the development of web applications that can be accessed from mobile devices.

This objective is described as follows:

From an initial case, suppose there is a web document that must be accessed from N devices with very diverse characteristics (See Figure 17)

What we want to avoid is developing N versions of that document, we are trying P number of versions to develop, where P<N. For example if an application for 100 devices is to be developed, (N=100) what is needed is the number of versions to be developed (P) with P<N (and theirs characteristics).

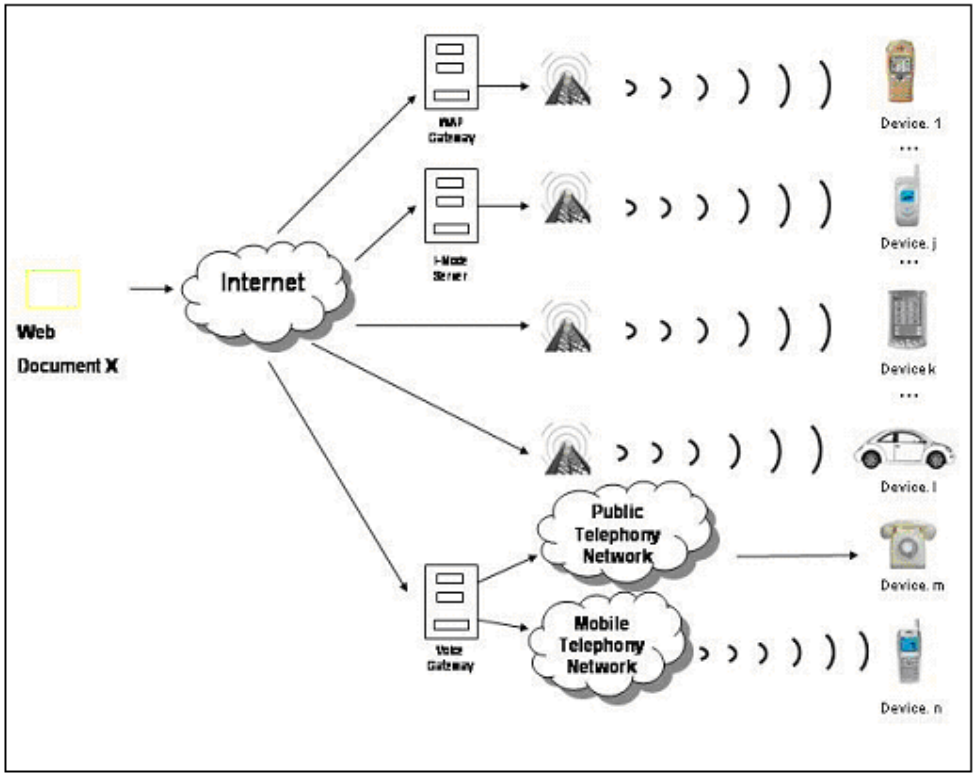


Figure 17. Multidevice Access to a Web document

In a more general case suppose that what is desired is the creation of an application that is accessed from N devices through the same URL: [See Figure 18]

http://wap.dgt.es/wap

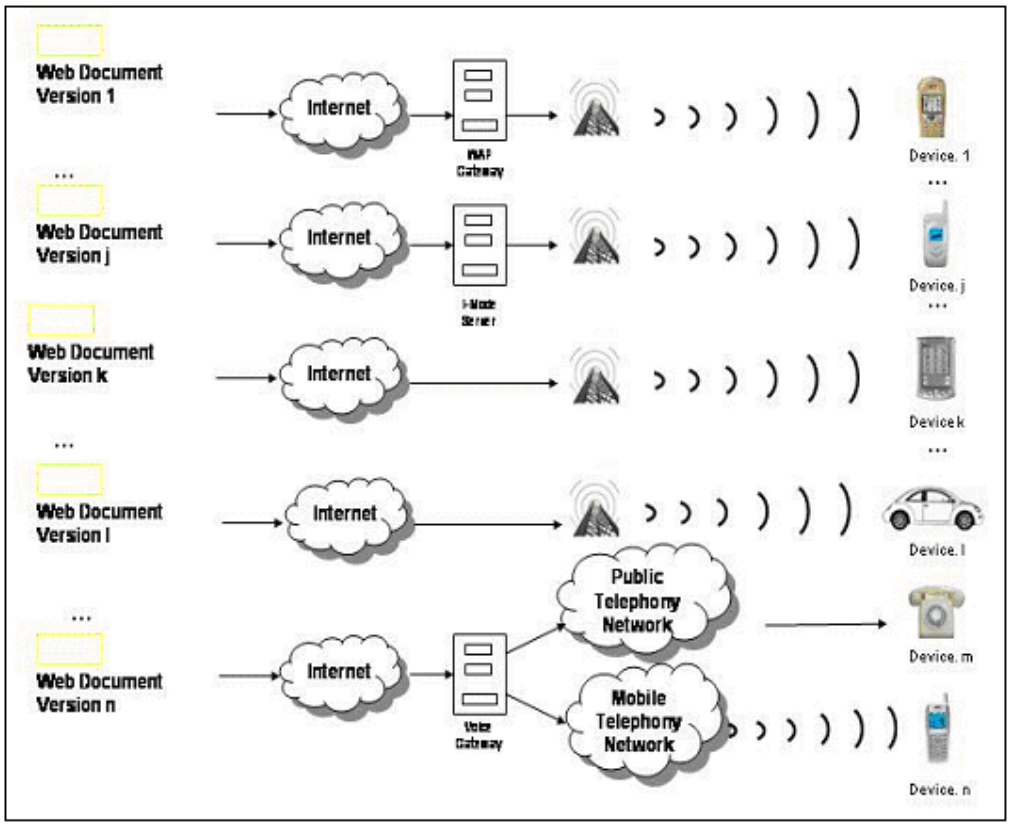


Figure 18. Multidevice Access to different Versions of a Web document

This is a more complex situation due to the presence of a hypertext structure whose design allow us to generate an interface that can be adapted in the best possible form to the capacities of the different devices.

The state of the art of the treatment of this problem were described in section I. In numeral 3.3. of this document the way in which the proposals of software engineering are evolving to include concepts related to the development of ubiquitous portals were described, mainly focused on the logical modelling of the application. Additionally the proposals of device modelling based on the Knowledge Representation area were described.

Elsewhere, in section 3.6.4.1 proposal Capability Classes was described, which poses the need of device groups for content generation and includes the possibility of defining these groups, which can be treated making use of the open-source library, DELI.

Also, existing commercial proposals were described, characterized by presenting solutions that are completely dependent on the manufacturer.

Once the existing proposals are analysed, the framework that is proposed as contribution in this thesis is described. In this sense the present proposal is based on the following contributions:

- A framework for multidevice web applications development that serves as guide not only to architects and software designers, but that also arrives to the application developer, by means of a bond between an approach for group characterization of existing devices, existing proposals for logical and conceptual modelling, and finally a generic architecture for the design of ubiquitous applications.

- A model of clustering or device grouping that allow us to guide the process of device group creation, in this way allowing the definition of an efficient mechanism of content development.

- An architecture for physical modelling of ubiquitous web applications that is independent of server-side platforms, scripting technologies and vendor solutions.

- An integrated computational tool that allows stylesheet editing for multi-device web applications based on device independence principles and script generation that allow the stylesheet transformation.

In this context, this chapter will describe this thesis and in the following one the contribution to the management of traffic information based on the raised thesis will be presented.

The framework is described in the following lines. The elements that compose them are as follows:

- Requirements.
- Device and target-user characterization.
- Logical and conceptual modelling.
- Architecture.
- Implementation and testing.

The flow that is associated to these stages can be seen in the Figure 19.

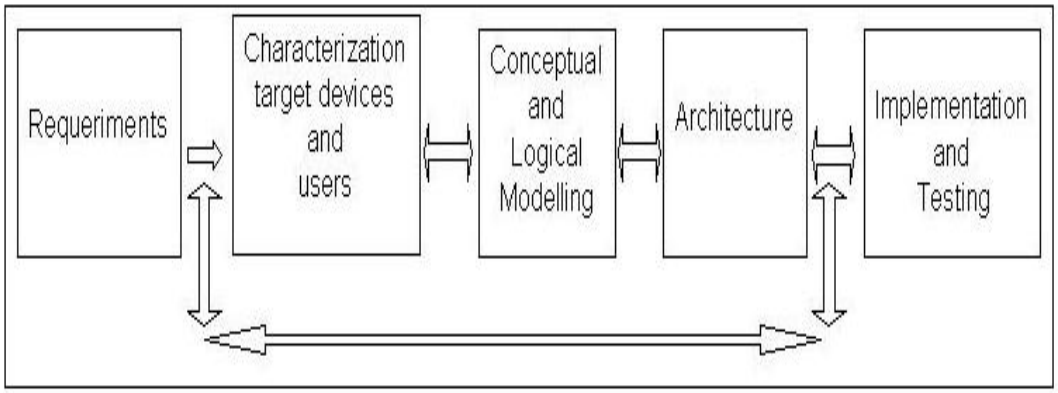


Figure 19. Framework for multi-device web applications development.

In the first stage, the planning process of the ubiquitous application and its requirements will be defined, emphasizing specific elements related to ubiquitous web applications as the following:

- Target devices and network technologies associated to them.
- Desired graphic and multimedia characteristics.
- Data that will be accessed (their structure and semantics will be defined later on) .

- Inclusion or not of file download or upload capabilities.
- Personalization levels based on location-based services and map management.
- Personalization levels for users or user groups.

Note: The last point will be discussed as an extension to the device characterization process, since in agreement with the problem boundary; it has been considered a future extension to this proposal.

The second stage is related to the grouping of devices by its capabilities and the corresponding mapping with the content that will be sent from the server. This process will be explained in chapter 5.

The third stage consists mainly of the application of techniques and existing methods developed by the software engineering communities for conceptual and logic modelling of web-site applications, based on the state of the art description made.

The fourth stage is based on the definition at physical modelling level of the architecture required for the application, for which in this proposal a generic framework is described. This architecture uses cluster modelling materialization as defined in the second stage

Finally, in the last stage the implementation and testing process are made. In this case the testing process is of special care, due to the difficulty in obtaining all the devices where the application will be executed, and to the existing limitations in the testing emulators.

Chapter 6 describes the architecture and implementation/testing staged.

The general process is based on a sequential flow, where the possibility of a revision of the previous phase exists. Once the phase of architecture is completed, a revision of the global process and specially the device characterization, before initiating the implementation process, is recommended.

Chapter 7 presents the integrated development environment for editing applications. It is based on the proposals for device independence.

The entire process raised in framework has been validated by means of the creation of diverse formats of the same data, and with the development of a multi-device and multilingual web application, whose main characteristics are described in chapters 8 and 9.

5. PROPOSAL FOR THE MAPPING OF DEVICE CAPABILITIES AND UBIQUITOUS WEB APPLICATIONS

5.1. SUMMARY

This chapter describes the proposal generated in this thesis in relation to the form in which a series of devices can be grouped, and the characteristics associated to those groups, from where it is possible to initiate a modelling and development process for an application based on different versions of device groups, and not by device versions.

This model has as input the data related to each of the devices and its properties for which web content is to be generated, and generates as output the web versions of the application, and its characteristics that should be developed.

Starting from this group it will be possible to continue with the process of logical and conceptual modelling, with the implementation of scripts and stylesheets that will interact with the purpose of generating the appropriate output for each device.

In the first part of the chapter, the data structure used in the proposed solution is described, and the grouping criteria is defined, based on the definition of property priorities and the way devices are grouped based on a difference percentages that are acceptable to the resolution values.

Afterwards a detailed description of the grouping process is made, which would be developed based on an example that uses the real data of 7 devices. In the same way, the form in which these devices should be added to the data structures associated to the proposal is presented.

Following, the algorithm that corresponds to the mechanism, as explained previously, is presented. Its efficiency is analysed by means of computational cost analysis, keeping in mind that this grouping is not carried out of execution time but rather as an activity in the final analysis phase.

Finally, possible extensions that could be carried out are described, which could be done to the generated proposal, which are based on the analysis of the relations that could exist between properties and the complexity associated to the multidimensional analysis of these possible relations.

Once the phase described in this chapter is finalized, a logical and conceptual modelling phase should continue, using some web software engineering technique that allows the handling of context and device characteristic treatment, and possibly user preference.

5.2 INTRODUCTION

This proposal is focused on the objective of obtaining a mechanism that permits systems to produce presentation data that can be rendered on a target device.

As we have analysed before, for presentation accessibility we have device-specific markup languages (e.g., PCs, and Pocket PCs support HTML, but Docomo mobile phones only support CHTML).

The presentation quality is affected by different screen sizes (e.g., PC screen size is above 800x600 pixels, Pocket PC screen size is around 320x240 pixels), all of them varying according to specific models.

Starting from the grouping process described as follows, a device grouping is to be carried out, in a way that will determine the number of groups to be created, and the characteristics of each group.

Starting from the group characterization, we can have an idea of the number of application versions that must be developed, and their characteristics.

To approach this problem the mapping technique looks for:

- First, find a number M of groups or clusters of devices defined by similar characteristics and priorities of design in relation to properties that are considered of greater relevance.

- Parting from the result obtained in this grouping process, it will be possible to have an initial idea of the number of applications to be developed, and the basic characteristics for each one of them.

- To identify devices that belongs to each group.

- To identify the minimum characteristics associated to each device group, in order to be able to create an appropriate mapping between the characteristics supported by each group of devices and the adapted content that will be sent from the server.

- Detailed study of the target devices as well as the basic characteristics that will be sent to each device, defining a single set of properties that will be used for that device (a markup language on which the contents will be developed, the image format that will be used, support or not of colours, etc.)

In this sense this approach will be optimised for these characteristics (in a way similar Web applications, where we can see in some web applications the message "Version optimised for X resolution").

This process of recommended characteristic definition would have to be accompanied by the contributions of at least a graphic designer or an interface expert.

A second element of the proposal is based on the definition of a tolerance index, which we will call *indtol* from now on, for the representation of information based on resolution information (wide high x).

A value of 0,1 in the index will indicate that it would be acceptable to display content of a device with height equal to 100 and width equal to 200 in one whose width is 110

and height 220, being a requisite that the value of tolerance will be adapted to the width as well as the height. This way, two devices that fulfil this characteristic could be grouped in a single cluster.

In formal terms the tolerance index corresponds to the following:

Indtol= (reference resolution – device resolution) / reference resolution

Where the reference resolution is that of a second device for which we are considering the possibility of presenting information. Therefore it constitutes a quantification of the resolution difference considered as acceptable for representing information.

5.3. DATA STRUCTURES

For the process of clustering and its characterization the use of two data structures is proposed: a hierarchic structure based on labelled trees and a multi-list.

- In the tree, levels correspond to the properties of interest of the devices, having as first level the markup language. The device groups would be added as tree leaves and they would always have an instance in the last tree level (each leaf in the last level of the tree will correspond to one device cluster). [See Figure 20]

  To determine the properties of each group, an ascendant route of the tree starting from any given group or a revision of the properties associated to each group can be made, in case these have been included as registers associated to each node of the multiconnected list.

- The linked list is maintained to have a device register that belongs to each group. Each node from the main list corresponds to a device group. At the same time, each node from the main list points to another list with the devices that belong to that group. (See Figure 21)

In the main list the maximum and minimum resolution within the group is additionally maintained, which is useful when a new device is to be added to each group.

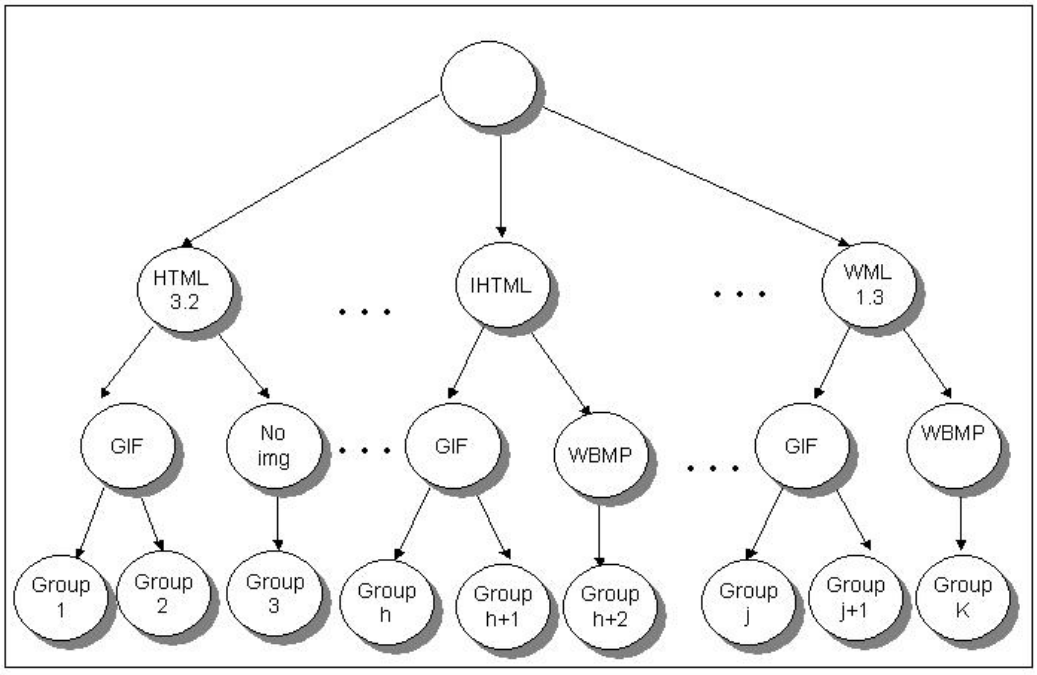


Figure 20. Cluster of Properties

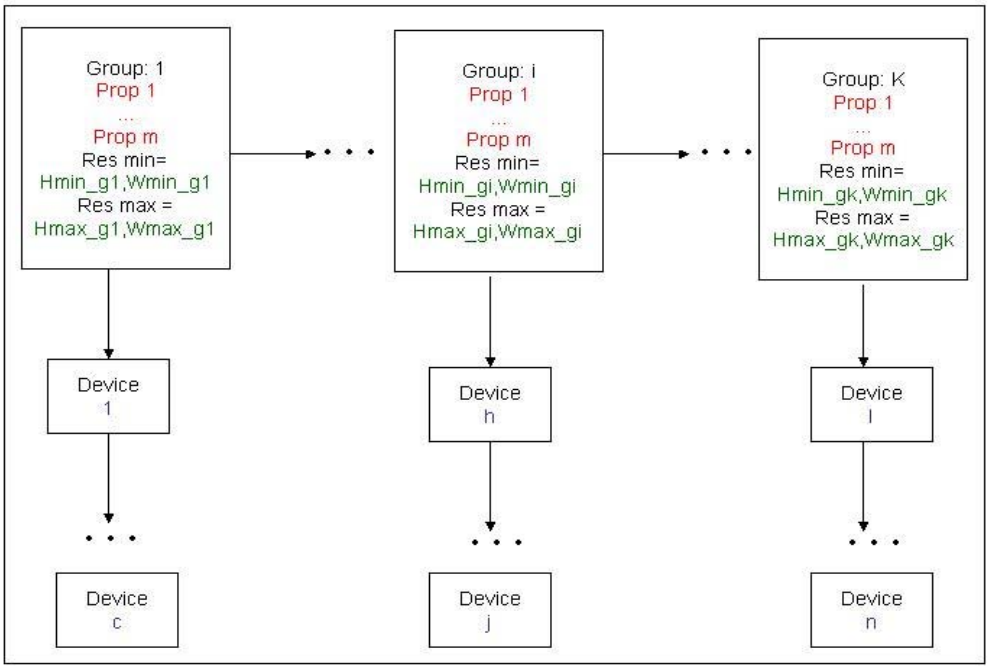


Figure 21. First Approach to the clustering multilist

5.4. GROUPING CRITERIA

In this method we propose the first tree level as markup language, since it is the starting point of a document editing process. Besides, the treatment of some device properties such as image format can depend on the markup language (for example, the inclusion of images in WBMP format always corresponds to documents written in WML). [4]

The resolution property will be taken as base for the comparisons with the minimum resolution of each group.

In the grouping mechanism that is proposed, there will be a minimum and maximum resolution value for each device group, which will be used to decide when a new device can be added to the group.

In order for a device with (h,w) resolution to be added to a group, the following possibilities exist:

When a new group is created in a sub-tree, the resolution of the first device to be added to the group will be registered as minimum and maximum group resolution.

If devices already exist in a group, there are the following possibilities of adding a device to a group:

Note: Hmax,Wmax are the greater heigh,width of the existing devices in a group, Hmin,Wmin are the smaller heigh,width of the existing devices in a group, and Hd,Wd are the heigh,width of the device to be added.

- Case #1: The height of a device is less than or equal to the maximum group width, and the width of the device is less than or equal to the maximum group height, but the width of the device must be greater than or equal to the minimum group width

[4] To reach this level of comprehension, the trials previous to establishing the framework as described in chapter 8 were important.

and the height of the device is greater than or equal to the minimum. This can be formally expressed as follows:

$$(Hd <= Hmax) \text{ and } (Wd <= Wmax)$$
$$\text{and}$$
$$(Hd >= Hmin) \text{ and } (Wd >= Wmin)$$

- Case #2: The width of a device is greater than (or equal to) the maximum group width, and the height of the device is greater than (or equal to) the maximum group height, but the width of the device must be greater than the minimum width, in a value equal to the percentage indicated by tolerance, and the height of the device is greater than the minimum height, in a value equal to the percentage indicated by the tolerance. This can be formally expressed as follows:

$$(Hd > Hmax) \text{ and } (Wd > Wmax)$$
$$\text{and}$$
$$(Hd <= (1+ indtol) * Hmin) \text{ and } (Wd <= (1+ indtol) * Wmin)$$

  In this case, the group values of maximum width and height must be updated, with the indicated values of width and height of the new device.

- Case #3: The width of a device is less than (or equal to) the minimum group width, and the height of the device is less than (or equal to) the minimum group height, but the maximum width must be greater than the width of the device in a value equal to the percentage indicated by the tolerance, and the maximum height must be greater that the height of the device in the same value as the percentage indicated by tolerance. This can be formally expressed as follows:

$$(Hd < Hmin) \text{ and } (Wd < Wmin)$$
$$\text{and}$$
$$(Hmax <= (1+ indtol) * Hd) \text{ and } (Wmax <= (1+ indtol) * Wd)$$

- In this case, the minimum values of width and height of the group must be updated, with the indicated values of width and height of the new device.

In case of there being more than one sub-group in which a device can be included, the criteria to be taken is to include the device in the group where there is a lesser difference in absolute value in relation to the maximum and minimum values, or human intervention could be considered to make the decision.

If the group was empty at the beginning, the H1,W1 resolution will be used to initialise the minimum and maximum group resolution. These values will have to be updated as the devices are added to each group.

Summarizing

- Each tree level represents an interest property to model.
- Each target device is added to the tree according to the priorities associated to it and to the tolerance index.
- The multilist is reviewed in order to determine which are the devices that compose each cluster.
- To determine the properties of each cluster, the tree levels are reviewed.

5.5 DESCRIPTION OF THE PROPOSAL BY EXAMPLE

Supposing that content is desired to be generated for the following devices: (See Table 3)

DEVICE	MARKUP LANGAUGE	IMAGE	SCREEN RES
1. Nec N21i (IMode)	IHTML-WML1.2	Gif/wbmp	120x130
2. Toshiba TS21i (IMode)	IHTML(only)	Gif/wbmp	128x121
3.SonyEricssonT68/R-201(WAP)	WML 1.3(1.2.1)	Gif/Jpeg/wbmp	101x80
4.Panasonic GD87 (WAP)	WML 1.3	Gif/Jpeg/wbmp	132x176
5.Sharp GX12 (WAP)	WML 1.3	Gif/Jpeg/wbmp	120x160
6.PDA IPAQ 3970	HTML 3.2	Gif/Jpeg	240x320
7.PDA T-Mobile MDA	HTML 3.2	Gif/Jpeg	240x320

Table 3. Device capabilities for the example

In the markup and IMAGE columns of table 3, what are considered the preferred values for these properties have been underlined in each of the devices. For example, in device #3, WML 1.3 and image format WBMP have been defined as preferred values. This decision is based on the analysis of the type of information required to use in the application and could be accompanied by the intervention of a graphic interfaces expert.

Starting from the first device on the list, Nec N2li, the analysis carried out to create an initial tree of devices is as follows:

- From the two markup languages that it supports, (IHTML, WML 1.2) IHTML has been defined as preferred value and this will be the one registered at the time the tree is created (this decision can depend on joint analysis of the device properties and can be accompanied by the criteria of an interface designer).

- From the image formats that it supports (GIF, WMP) the GIF format has been determined as preferred value.

Following a process similar to the other devices, and adding the last level of resolution, the tree that would initially be created, without realizing any type of grouping, can be visualized in the Figure 22.

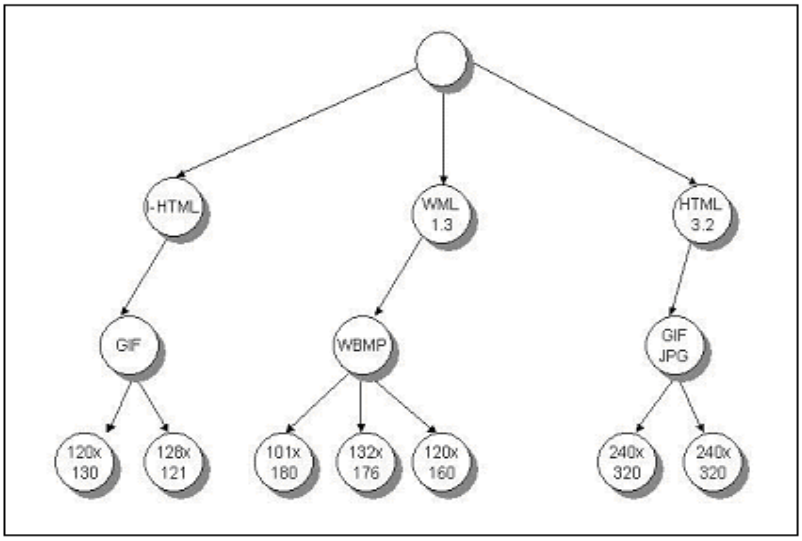


Figure 22. Initial tree with devices

Following, the tree creation process of the grouping, the associated multi-list that allows group definition will be described.

To add the first device, the starting point is the inclusion of the preferred markup language, which according to Table 2, corresponds to IHTML. Afterwards, in the following level a child node corresponding to preferred image format GIF is added.

If other properties of interest exist, they would in turn be added as child nodes, adding a property to each tree level. Finally an identifier corresponding to the first group would be added, since this node would form a first group. (See Figure 23)

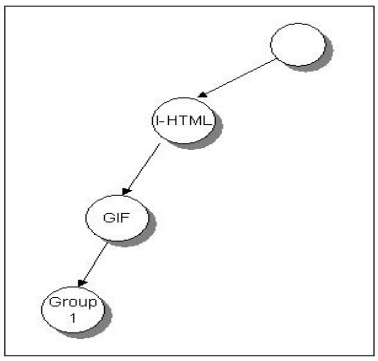


Figure 23. Tree with the first group

Additionally, a first node in the multi-list is created, with the first group identifier as well as the initialisation of the minimum and maximum group resolution, with its corresponding value. (See Figure 24)

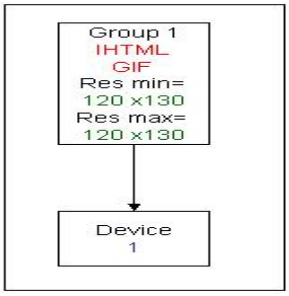


Figure 24. List with the first device

To add the second node the process is similar, in this case the properties are IHTML, GIF image format and 128x121 resolution. The corresponding tree can be seen in the figure 25.

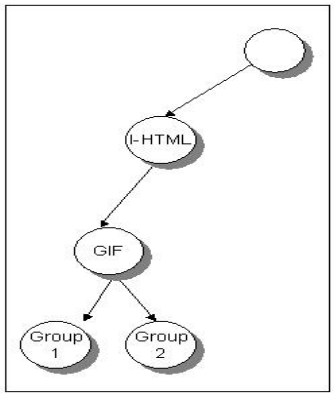


Figure 25. Tree with the second group

Keeping in mind that the resolution corresponding to this device does not fulfil the criteria defined to add a node to the only existing group in the last sub-tree, a new node in the multilist is to be created. (See Figure 26)

It is noticeable how, regardless of dimension w, belonging to the resolution (h,w), having a value of 121, which is less than the maximum value of the first group (wmax = 130), the first dimension h with a value of 128, could not be represented within the maximum group dimension hmax = 120 (criteria 1 is not fulfilled). Therefore this device could not be added to this group.

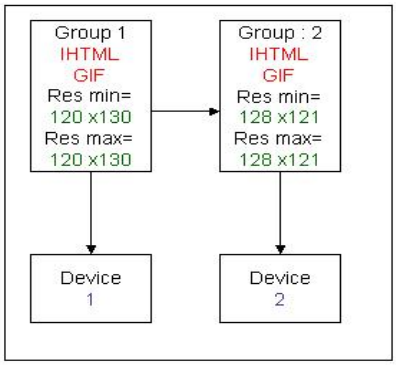


Figure 26. List with the second device

The third device is added to the tree and to the list according to the values of its high preferred properties; markup WML 1.3, image format WBMP and 128x80 resolution. (See Figure 27)

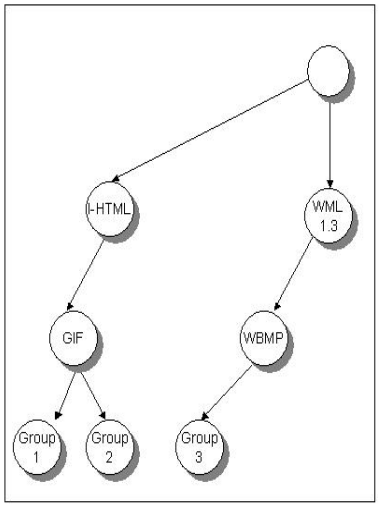


Figure 27. Tree with the third group

Again to add a device to the multi-list, a new group is created, since in the subtree markup language WML 1.3 there was no existing subgroup. (See Figure 28)

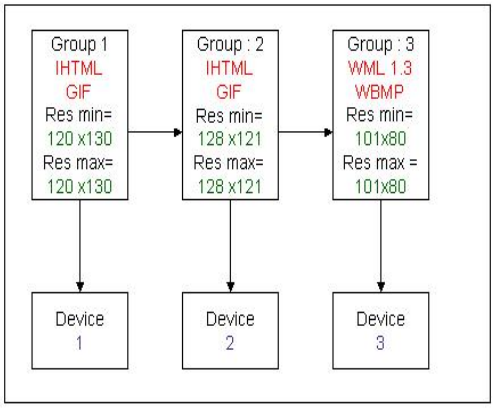


Figure 28. List with the third group

The properties corresponding to the fourth device are: markup WML 1.3, image format GIF and 132x176 resolution (See Figure 29). It can be observed how the resolution of this device cannot be represented by the existing resolution within the existing group in the sub-tree based on markup WML 1.3 and WBMP format. (The maximum and minimum resolution of subgroup is 101x80).

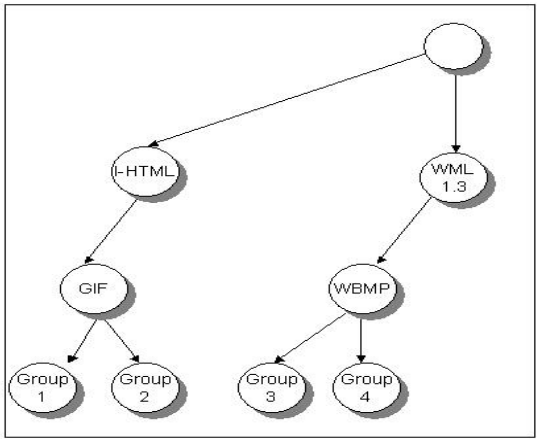


Figure 29. Tree with the fourth group

Therefore, this device will correspond to a new node in the multilist. (See Figure 30)

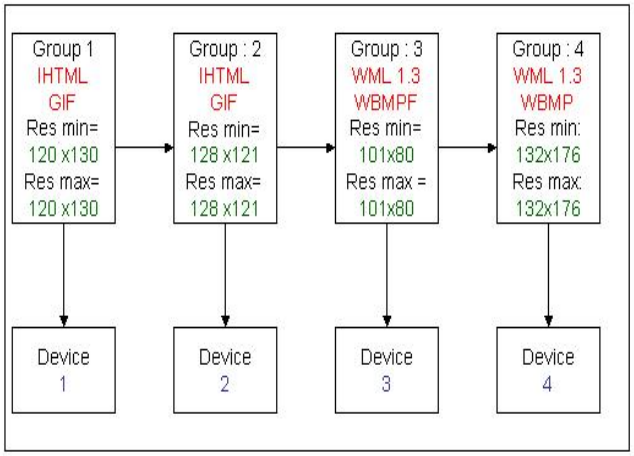


Figure 30. List with the fourth group

The properties of device number 5 are: WML 1.3 markup, image format GIF and 120x160 resolution. To add it, these are taken as the comparison basis for sub-list groups corresponding to WML 1.3/GIF.

Group number 3 has a minimum and maximum resolution of 101x80, with which the resolution of device number 5 (120x160) could not be represented in this group.

Group number 4 has a minimum and maximum resolution of 132x176, with which the resolution of device number 5 (120x160) could be represented in this group. In this case the node must be added to this list and the value of minimum resolution must be updated. (See Figure 31)

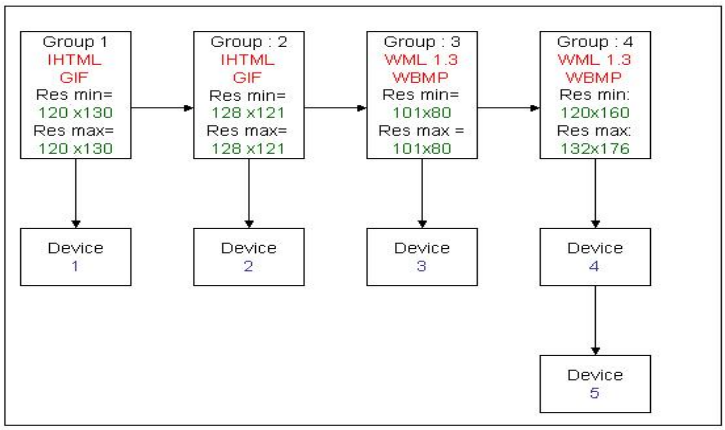


Figure 31. List with the fifth group

The sixth device is added parting from the following properties, markup HTML 3.2, image format GIF and 240x320 resolution. The corresponding tree representation can be seen in figure 32.

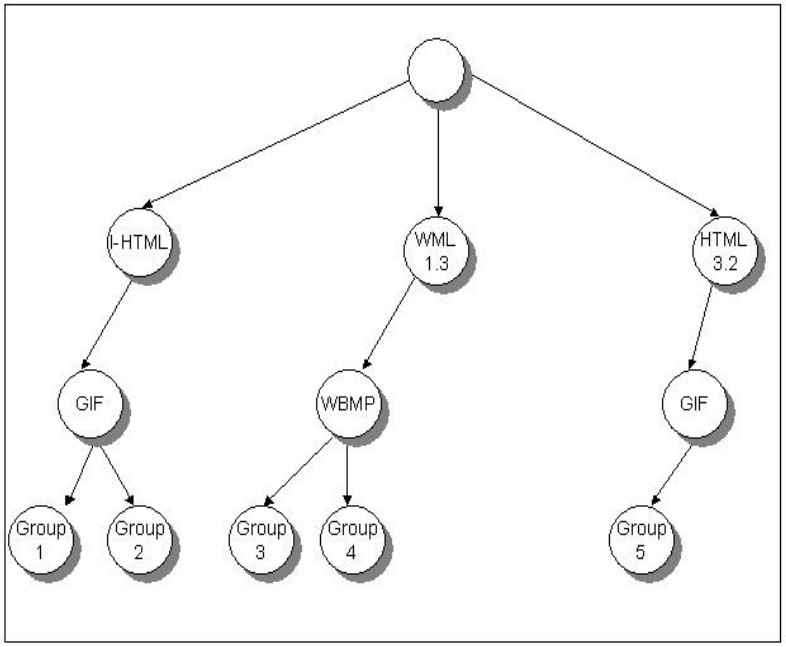


Figure 32. Tree with the fifth group

Keeping in mind that the new device generates a new group, the corresponding representation in the multi-list is as follows. (See Figure 33)

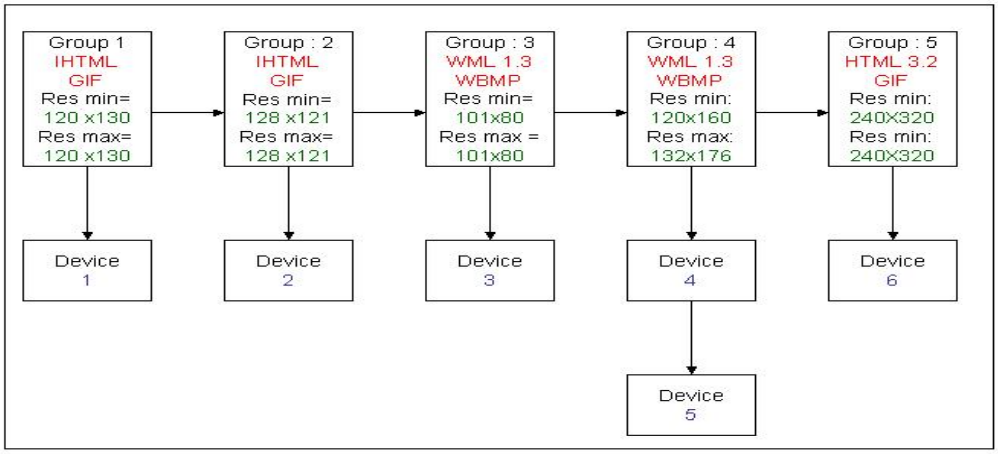


Figure 33. List with the fifth group and sixth device

The properties corresponding to the seventh device are: HTML 3.2, GIF and 240x320 resolution, with which a new tree node is not generated, since a leaf with identical properties exists. When the device is included to the last group, the multi-list representation corresponds to what is described in figure 34.

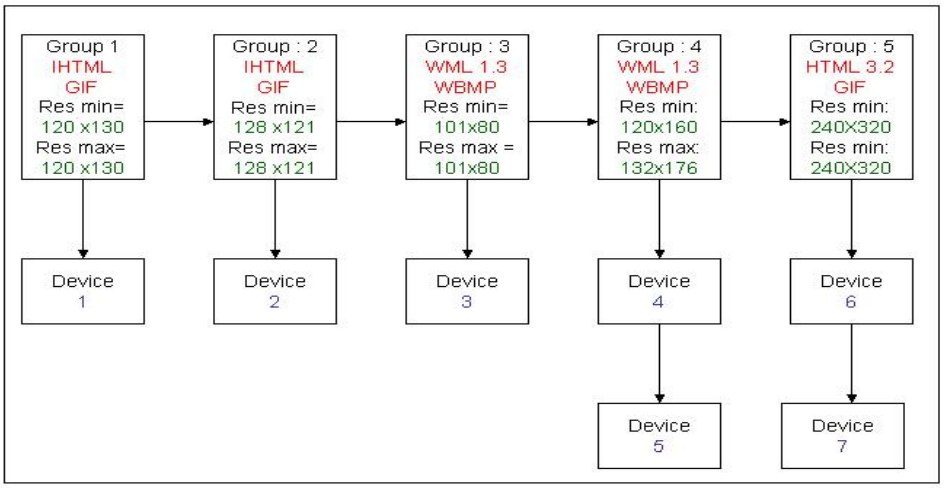


Figure 34. List with the fifth group and seventh device

And the final tree representation can be observed in figure 35.

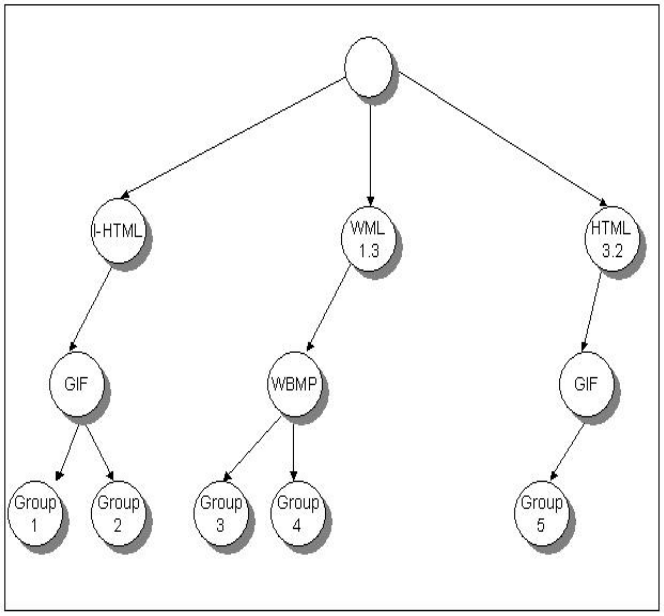


Figure 35. Final representation of the tree

Carrying out a comparison with the initial tree representation (see Figure 22) and parting from the observation of the tree in figure 35, we can see that the number of groups is 5, which indicates that 5 new versions of the application should be created for the 7 devices.

In this way, the objective of counting on a number of versions less than the number of devices is reached.

The characteristics of each version could be obtained by tree route starting at the rode node until reach each group, or consulting the properties registered in each subgroup in case these values in each node were stored.

5.6. ALGORITHM FORMALIZATION

Having N devices with M properties, where M is the property corresponding to the device resolution, the algorithm to add each one of the devices to the tree is the following one:

Input
Hd, Wd (Height and Width of K-th device to be added)

Begin
If tree is empty *then*
Create root node
Add the first device as follows:
Add a node in each level of the tree (one node for each property)
Add the first node to the multilinked list and associate the device to the first group
else
'Having the k-th device to be added:
Control =0
level = 1
while level <= M-1
*If (*Control node has not a child with a value of the property equals to the value

```
            of   property of this level) then
                 Add a child of Control to he current property
                 Control=added node
            else
               Control=current node
        End-if
        Level = Level + 1
    end-while

    if sub-tree in control is empty  (subtree of resolution) then
        Add a new group as child of  Control
        Add to the multilist the properties of the devices and its identifier
    Else
      ' P is Number of child groups of a control node
      flag_add=0
      i=1
     dif_min=10.000 (some high value)
      while  ( I <= p )
        If   (   (Hd  <=  Hmaxx )   and     (Wd <=  Wmaxx)
                              and
              ( Hd  >=  Hminx )  and    (Wd >=  Wminx)     )   then
                        dif_min  = abs (Hd – Hminx) + abs (Wd – Wminx)
.                        if (dif <dif_min) then
                                dif_min=dif
                                selgruop=I
                         end if
                        flag_add=1
              else

            If   (   (Hd  >  Hmaxx )   and     (Wd >  Wmaxx)
                              and
             ( Hd  < = (1 + idtol)*  Hminx )  and    ( Wd <= (1+idtol)* Wminx)     )
              then
                        dif  = abs (Hd – Hminx) + abs (Wd – Wminx)
```

```
                    if (dif <dif_min) then
                            dif_min=dif
                            selgruop=i
                    end if
                  flag_add=1
          else
            If   (   (Hd  <  Hminx )   and     (Wd <  Wminx)   )
                        and
              ( Hmaxx  <= (1 + idtol)*  Hd )  and    ( Wmaxx <=  (1+idtol)* Wd)     )
                 then
                    dif  = abs (Hmaxx – Hd) + abs (Wmaxx – Wd)
.                    if (dif <dif_min) then
                            dif_min=dif
                            selgruop=i
                    end if
                  flag_add=1

              else

           end-if
        end-if
      end-if
      i=i+1
      end-while

      If flag _add=0 then
              Add a new group to the tree
              Add a new node in the main list
              Add the new device as child of the new node in the list
        else
              Add the new device as child of the group selgroup
      end-if

    End-if 'if  subtree in control is empty
```

End-if 'if tree is empty
End

Variables
M: Number of properties by device
Level: Value of the property of the device corresponding to any given level
Control: Used to determine the node that is being processed.
flag_add: Used to define that a device was added to the multilist.
p: Number of child groups of a control node
i: Used to go over each of the existing groups in the last level at any given time
Hminx, Wminx : the minimal resolution of each of P groups that are the children of Control
Hmaxx, Wmaxx: the maximun resolutions of each of P groups that are the children of Control

5.6.1. Computational cost

In the process of creating the tree and the list, the computational cost for each device with M properties is O (M), and for all of the devices is O(N*M), since it is necessary to process the N devices to add them to the tree and list as nodes.

For the determination process of properties or group characteristics, two options exist:

- For each one of the existing groups of the tree, realize a process by levels of the tree until the root is reached. In this case the computational cost has a direct relation with the number of tree levels (interest properties) and the computational complexity would be of O (N*M) .

- If storage of the properties of each group there are registers associated to each element of the main list to be made, this would avoid making a search by levels and there would be direct access to the properties of a group by means of its identifier.

This computational cost is not in run time, since the grouping is done as a previous step to the development.

Additional Considerations

Before initiating the grouping process, the possibility of realizing a refinement starting from the initial tree of properties may be considered.

In this case, a tree without defined groups would be created, and including only the value of resolution as the last group level. Subsequently sorting the data following a predefined criteria (height, width, etc) can be realized. In this case we would introduce the computational cost of the sorting process (O (n*log(n) using a efficient sorting algorithm))
Once the organization is carried out, the grouping of devices can be achieved. This way, different perspectives of grouping can be obtained, but the computational cost would be higher.

In any case, starting from any criteria that are to be employed, human intervention can be considered with the purpose of improving the results obtained.

5.7. EXTENSIONS

5.7.1. Partial belonging to multiple groups

With the first approach that has been explained previously, each device belongs to only one group and they have considered within each level of the tree the occurrence of a single property within each device (for example to give single support to a format of images, within x formats supported by the device).

This level of property is summarized in the table number 4, where each device belongs to a single group (1) and in the others it has a level of property of 0.

Device	Group 1	Group 2	Group3
D1	1	0	0
D2	0	1	0
D3	1	0	0
D4	0	0	1
D5	0	1	0
D6	0	0	0

Table 4. Devices in a unique group

In this new approach, there is the possibility of each device set belonging partially to different groups (the sum of the property levels will have to be 1)

A first option is based on quantifying the level of a property within each level, using an equitable valuation between the possible existing values of a property. For inferior levels the process starting from the existing valuation in the node corresponding to the previous level would be repeated.

Next, we describe an example that illustrates this process:

If one a device supports XHTMLMP, WML 1,3, iHTML, and has determined to give a 60% of preference the use of XHTMLMP in that device and that the others would have an equal preferred percentage, equivalent to 20% each one.

For the second level, if the device support GIF and png formats in XHTMLMP and IHTML; but only WBMP format in WML 1.3. In this case, the property of the device to each group would be the following one:

For the case of a group of devices the situation would be as follows: (See Table 5)

XHTMLMP+ GIF	XHTMLMP+ PNG	WML13+ WBMP	iHTML+ GIF	IHTML+ PNG
0.30	0.30	0.20	0.10	0.10

Table 5. Devices in multiple groups

Finally, the criteria for the definition that represents the most appropriate option can be taken from different options: as the maxim property, as anyone of the two higher belonging, or a decision supported by the human intervention

5.7.2. Relations among properties: Self Organizing MAPS (SOMs)

We can try to consider in the model aspects to give greater priority to a property than to another one, and making variations of these interesting properties, using approaches such as Self Organizing Maps by Kohonen [Hon04]. It would be also possible to find different groups that represent, in a better form, the interest by specific properties. [See Figure 36]

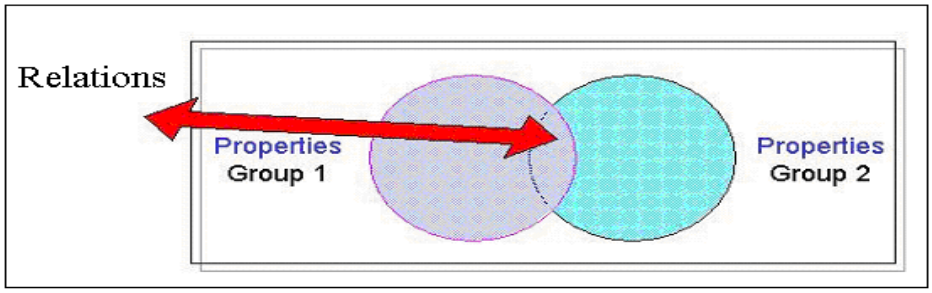


Figure 36. Relations among properties:

If the functional dependency analysis exists between device properties, we can consider the situations as described below.

Supposing that the existing dependency between image format I, in relation to language L, is to be given a model, said relation could be expressed by a function F, as follows:

$$I = F (L)$$

Starting from this dependency function, a certain type of group could be defined, which could be graphically expressed as in the figure 37.

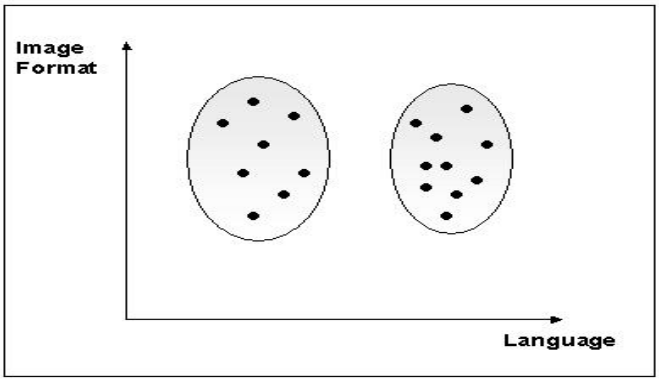


Figure 37. Lineal relation between properties

If we take into account the variation of this dependency with time, the relationship can be expressed by a differential equation.

$$\partial I / \partial T = \partial F(L) / \partial T$$

If we now consider the existence of a third property, such as colour 'C', the graphic representation of a possible grouping could be expressed as in Figure 38.

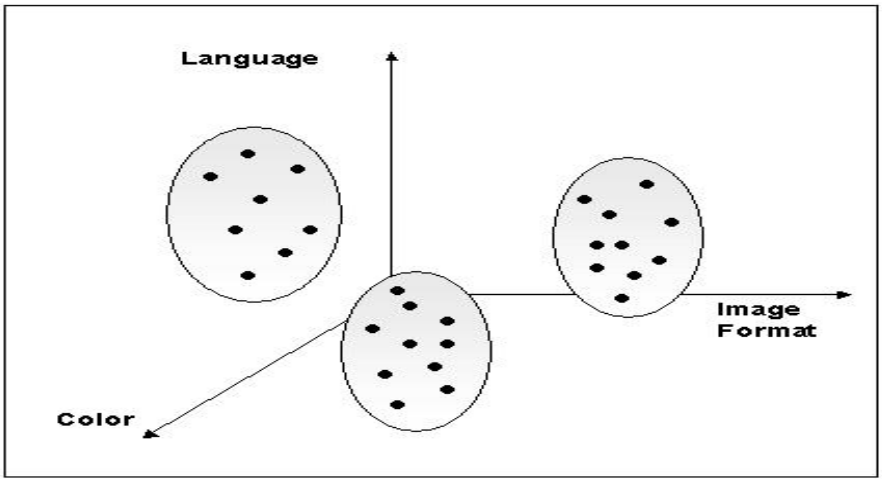


Figure 38. Multidimensional relation among properties

And if we consider a new time variation in this dependency, the relation could be expressed as follows:

$$\partial I / \partial T = G (\partial F(L) / \partial T , \partial F(C) / \partial T)$$

If we finally consider the existence of k properties:

$P = \{ p_1, p_2, p_3, ..., p_k \}$

and if the variation of the dependency between properties is the time function, we find a system of temporal non-linear differential equations.

The solution for this multi-dimensional systems is complex, there are only numerical approximations based on simplifications to model them.

Due to what has been exposed, the approximation to define a grouping model described in this chapter constitutes a linearization based on "property priorities" that allows us to simplify the treatment of the problem.

5.8. CONCLUSIONS

The only similar proposal is the capability classes by Mark Butler of HP. In this proposal cluster logics are defined from a series of operations that allow the grouping of the devices, although an explicit recommendation is not made of the process follow to characterize clusters. The existing commercial solutions make certain type of adaptation, but their operation is known in form of black box.

Based on the mapping described before, at design time an abstract model to manage the markup languages is provided to authors to build a device-independent presentation model. At runtime, all tags will be interpreted as a set of device-specific tags, based on the device's supported markup language. The supported markup language information is one of the delivery contexts that we assume to be retrieved from the CC/PP protocol.

6. ARCHITECTURE, IMPLEMENTATION AND TESTING STAGES

6.1 SUMMARY

This chapter describes the framework stage that follows the logical and conceptual application modelling process.

Once the previous phase is carried out, the physical modelling process must be done, for which an architecture that can be used as guide for developments that employ it, or for other architectures that can expand on it, are described in this proposal.

Initially the layers of the proposed architecture are described, and afterwards the implementation process is described as well.

Afterwards, different instances of this architecture are described, which serve to show the independence of server solutions and proposed model technologies. Additionally, the result of the grouping process that was described in the previous chapter can be incorporated into this architecture.

The configuration that should be defined in the server to associate the device groups to URLs is described. This group treatment can also be handled starting from a single server script that receives some type of parameter, and allows the realization of the required transformation processes.

Additionally, a functional view is presented, which serves to illustrate the architectural function in a real server, based on the interaction between WAP and Imode devices when a mobile application is accessed.

In the final part of this chapter, the implementation stages and testing as part of the framework, are described.

6.2. INTRODUCTION

This architecture is based on an extension of the three-layer model typical of Web applications, but also includes the required elements for device treatment, as well as a consideration of the semantic treatment guided by the proposals defined in chapter 3.

Initially, the layers that form the architecture are described: data layer, logical layer and presentation layer, without going into detail in relation to implementation processes that are described afterwards.

Finally the different implementation activities that must be carried out in each layer are listed, as well as the special considerations that must be kept in mind in the ubiquitous portal testing process. We indicate the different components that must be tested and the way in which these tests must be developed.

6.3. THE ARCHITECTURE

The proposed architecture for the development of ubiquitous web application is based on 3 main layers (See Figure 39):

- Data Layer
- Logic Layer
- Presentation layer

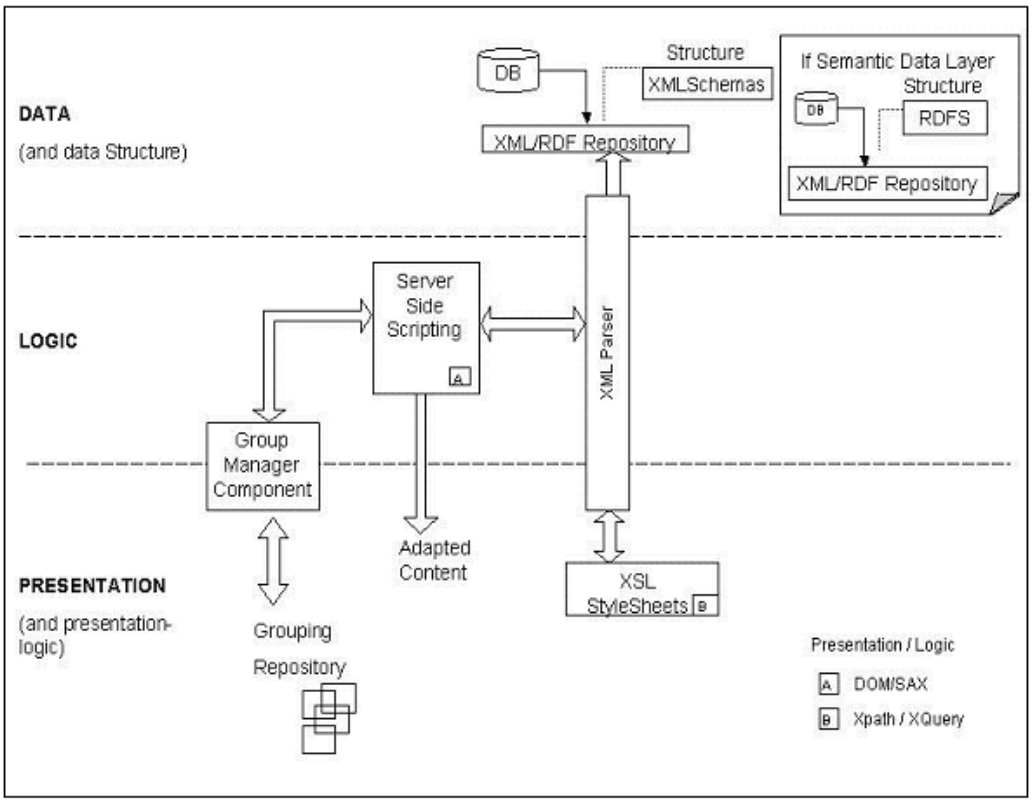


Figure 39. Architecture for multidevice web applications development

6.3.1. Layers of the architecture

6.3.1.1. Data layer

The data layer corresponds to the data repository, which is represented by means of XML documents. In the case of an existing database system a representation in XML would be needed for the later treatment by means of transformations that allow content to be generated in the required markup language.

Within this layer there are also the components that define the structure of these data, for which the use of XML schemes allow more possibilities than the DTD (Document type definition) .

Additionally, if it is planned to develop a web application based on intelligent representation of knowledge, we probable need to define a semantic layer, with a mechanism of representation of the knowledge, this would be probably be based on

RDF representations and their corresponding structures of data, or on some other language defines a semantic model, like DAM+OIL(DARPA Agent Markup Language+Ontology Interchange Language) and OWL (Web Ontology Language).

The importance of the inclusion of semantic capabilities lies in the possibility of counting with a level of abstraction superior to that of the markup. In other words, the simple definition of tags is extended; for example, the road tag is extended to define the highway where an incidence exists.

When a semantic model is included, a vocabulary that is expressed by means of an ontology must be proposed, as well as axioms and rules associated to this definition. Starting from this semantic model, it is possible to realize subsequent treatment by means of software components based on artificial intelligence that will interact with the ontology.

6.3.1.2. Logic layer

In the logical layer, the connection between the data and the presentation layers takes place, this can be developed using diverse technologies like Java Servlets, scripts based on JSP (Java Server Pages), PHP, ASP (Active Server Pages), or developments based on Microsoft Dot Net where a total control on the transformation processes is required. As a consequence of this, the proposed architecture does not depend on any of these technologies

For access to XML data a XML parser is used, it enables the script processing engine to manage the process of data and stylesheet transformations that define the format, to generate content adapted to the capabilities of a particular device.

The transformation of XML data is done jointly by the script engine and the selected parser through XSLT, and with the intervention of a component from the processing of groups.

A fundamental module present in the logical layer is the management of groups, there will have to be a component or library that manages the differences between the existing group devices, and access to its properties.

By means of previously established groupings, the groups of devices with common characteristics and their characteristics are detected, optimising in this form the documents to be developed

The group management component will allow access to objects and methods that permit the access to device properties that make a request, for example allowing access to the markup language, to supported image formats, etc. This group treatment must be included within the scripts responsible for negotiating content and transformations.

6.3.1.3. Presentation layer

Finally, the presentation layer contains the stylesheets with the format associated to the different groups from devices determined in the analysis phase, as well as the description of the properties of the different groups

In the style sheets the reference to actual elements becomes a determined XML document by means of languages such as XPath or XQuery, references to multiple documents are also made by means of XInclude. In exceptional cases, it is possible that some type of processing by means of DOM (Document Object Model) or SAX (Simple for API XML) within scripts is made using the selected processing engine.

From the tests made on the implementation processes we detected that due to the impossibility of defining variables in the style sheets, it was necessary to use a combination of techniques for the processing of XML trees already mentioned with dynamic generation of interface elements.

The presentation layer is the repository of clustering, which communicates with the grouping manager to detect the most appropriate cluster for each device.

Based on the mapping described before, at design time an abstract model to manage the markup languages is provided to authors to build a device-independent presentation model.

At runtime, all markups will be interpreted as a set of device-specific markups, based on the device's supported markup language. The supported markup language information is one of the delivery contexts that we assume to be retrieved from the CC/PP protocol.

6.3.4. Implementation of The Architecture

6.3.4.1. Experiments: Multiple Instances of the architecture

In order to verify the extensibility and applicability of the platform, diverse tests have been made. They have been applied to different server configurations. Although an XML parser could be developed, it has been included in the reference to existing solutions.

- Configuration based on Java as server-side scripting technology. In this case Xalan has been selected as XML parser. (See Figure 40).

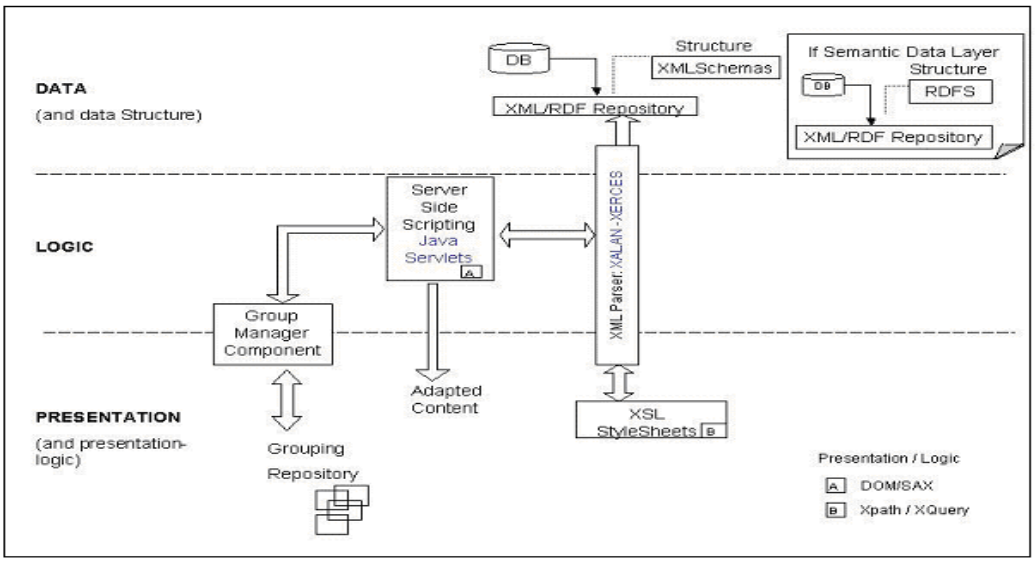


Figure 40. Instance of Architecture base on Java scripts

- Configuration based on PHP as server-side scripting technology. In this case Sablotron has been selected as XML parser. (See Figure 41)

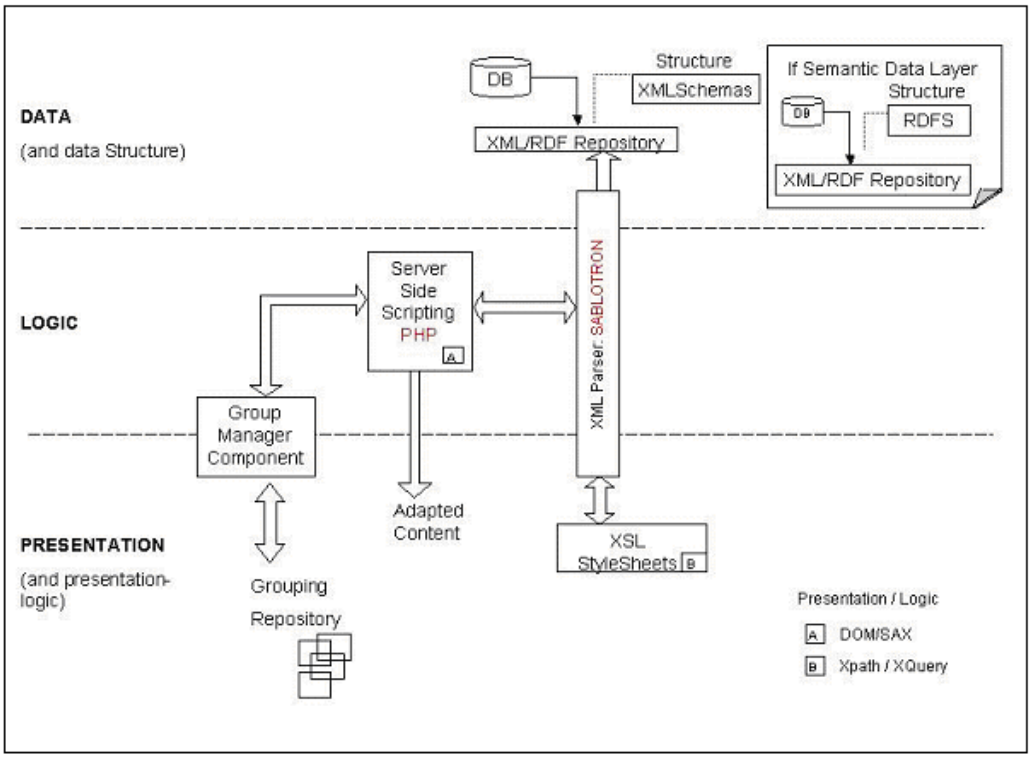


Figure 41. Instance of Architecture base on PHP scripts

- Configuration based on ASP as server-side scripting technology. (See Figure 42)

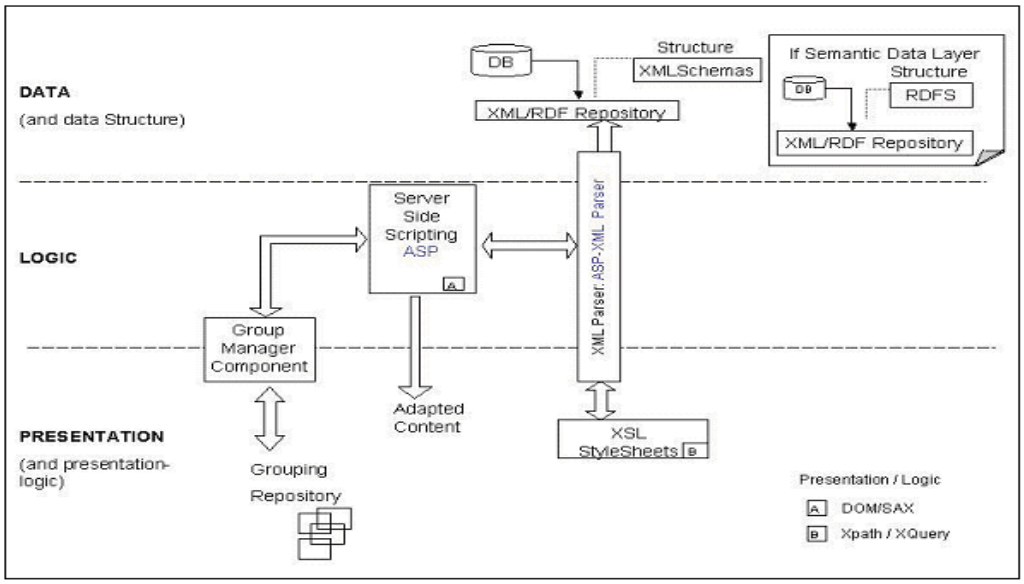


Figure 42. Instance of Architecture base on ASP scripts

6.3.4.2. Functional View

The functional implementation of this architecture is based on an extension to the processing of a user agent that identifies the client who makes a request, and it can also be based on any CC/PP implementation that facilitates the detection of the device capabilities. Figure 43 shows two requests made from two devices based on different technologies

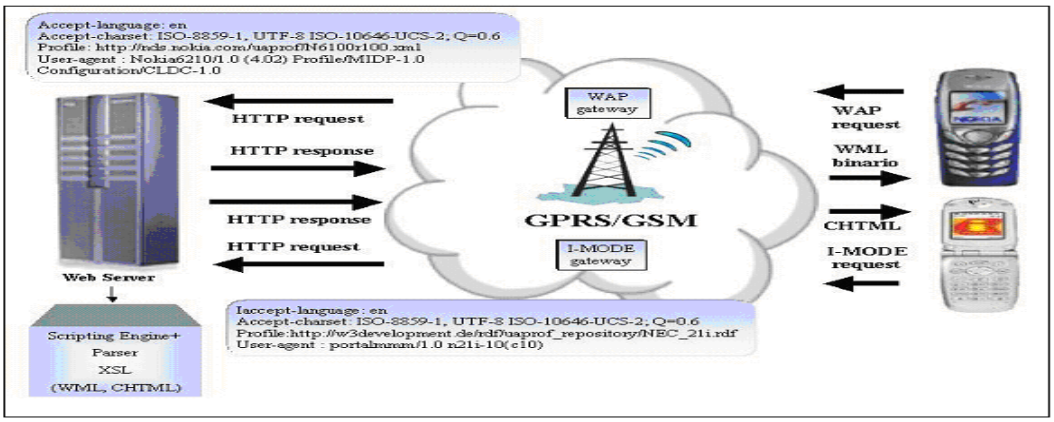


Figure 43. Basic View of the Operation of the architecture

When a device accesses the server by means of an HTTP request using the user-agent. The received user-agent is verified in a file that contains different user-agent and the corresponding group is found. The identifier of this group is given back to the server, with the most appropriate format that has been determined, and the corresponding stylesheet is applied. The result is transformed by means of the scripting engine in the server and the content is sent to the user. (See Figure 44)

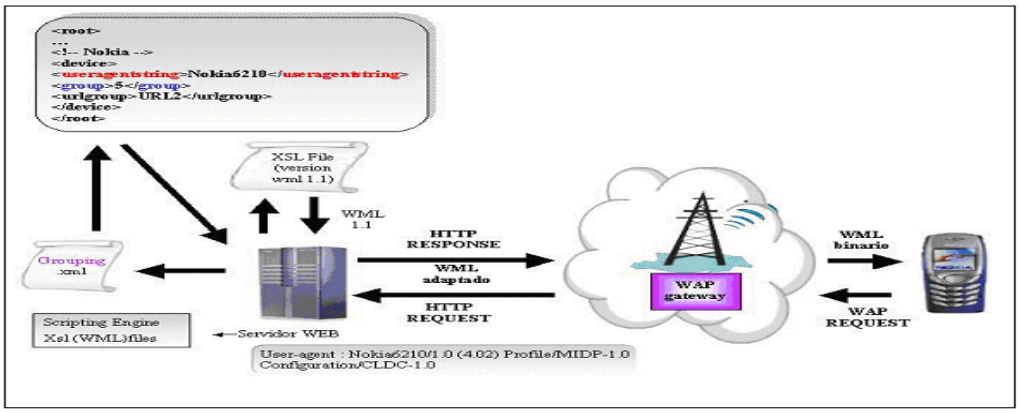


Figure 44. Extended View of the Operation of the architecture

With the results of the model of clustering used a configuration file will be created reflecting the result of the grouping. (See Table 6)

Device	User-agent	Group	URL group
1. Nec N21i (IMode)	portalmmm/1.0 n21i-10(c10)	1	URLg1
2. Toshiba TS21i (IMode)	portalmmm/1.0 TS21iC-10(c10)	2	URLg2
3.SonyEricssonT68/ R-201(WAP)	SonyEricssonT68/R-201A UP.Link/5.1.1.5	3	URLg3
4.Panasonic GD87 (WAP)	Panasonic-GAD87/A19	5	URLg4
5.Sharp GX12 (WAP)	SHARP-TQ-GX12/1.0 Profile/MIDP-1.0 Configuration/CLDC-1.0 UP.Browser/6.1.0.5.119 (GUI) MMP/1.0	4	URLg4
6.PDA IPAQ 3970	Mozilla/4.0 (compatible; MSIE 4.01; Windows CE; PPC; 240x320)	5	URLg5
7.PDA T-Mobile MDA	Mozilla/2.0 (compatible; MSIE 3.02; Windows CE; PPC; 240x320)	5	URLg5

Table 6. URL for groups of devices

6.3.5. Comparison with other proposals

Some of the advantages that the proposed architecture offers include the following:

- It can serve as base for the development of multidevice web applications and complement existing proposals of logical modelling or as base for a new modelling proposal.

- The independence in relation to vendor solutions, allows a migration to different server systems from server.

- It uses the proposals developed by the international community and formed in the recommendations of the W3C.

6.4. IMPLEMENTATION STAGE OF THE FRAMEWORK

Once the stages previous to the implementation process are realized, according to what is defined in figure 45, the procedure will be as follows.

Before initiating this phase, the existence of the following products from previous stages must be verified:

- Analysis phase results.
- Device grouping and group characteristics model.
- Logical and conceptual modelling of the application.
- Group navigation modelling.

Starting from the requirements analysis, the following aspects must be defined:

- Select the server's operating system.
- Select the script technology.
- Select the Web server and script-processing engine.
- Select the XML parser.
- Select or develop the device capability-processing library.

Afterwards, the following implementation activities must be carried out:

- Data layer and/or semantics implementation
  - Data structure definition.
  - Ontology definition in case of including a semantic model.
  - Creation of XML and/or RDF files.
  - Creation of data files in XML format or definition of intermediate representation mechanisms.
- Presentation layer implementation

- Obtaining device profiles (starting from manufacturer data or from existing repositories).
- Creation of grouping files.
- Creation of functional group interface view.
- Creation of stylesheets for the different device groups.

- Logic layer implementation

  - Development and/or generation of the scripts for processing device capacity.
  - In case of an existing semantic layer, development of software components in charge of interaction with the ontology that has been defined.
  - Development of scripts for stylesheet transformation.

6.5. TESTING STAGE OF THE FRAMEWORK

The testing process of ubiquitous applications differs from regular function testing used for any software product.

Besides the many complex technologies involved in a pervasive Internet application, there are specific conditions of the mobile environments that affect testing. Among others, some of these conditions are:

- Typing on a wireless device is not as easy as on a full-size screen.

- The screen of all target devices can display the content.

- The cost of surfing on mobile networks.

- The support for international characters in different devices.

Getting the testing feedback in an early phase can improve the end performance of the final product.

It is necessary to perform tests by using emulators from the start of the development process and perform this test on real devices and networks as products mature.

It is also necessary to run tests on real platforms, since the performance of emulators and real devices often differs significantly.

With an emulator, it is easy and cheap to record a test scenario and then repeat that scenario in order to determine whether the developer has managed to improve the performance.

The testing of wireless applications must include:

- ✓ *Interface testing*

This stage involves making sure that the application works in a good way on a wide number of target devices and that it offers a user-friendly interface.

The most common way for this is to concentrate on testing the functionality and user interfaces offline, before starting to optimise the wireless properties.

This part of the testing should also include usability tests.

- ✓ *Network performance testing:*

This test involves making sure that the application performs well even in the more difficult conditions, such as changing bandwidth or having interruptions due to absence of coverage.

Even when the network is available, it is difficult to use the network for more than a final end-to-end test. With a real network, you cannot repeat the exact test scenario twice because of the many factors that affect the performance of a real network.

The network testing can be made using a network laboratory or a network-testing tool such as Global Applications Test Environment (GATE) available at Mobile Applications initiative (www.mobileapplicationsinitiative.com)

GATE was initially developed as a GPRS test environment but now includes 3G networks technologies. GATE is connected to the application where the live network has been, and emulates it.

- ✓ *Server-side testing.*

This is needed to test both the functionality and robustness of the server. We also need to be sure that the application works on different server platforms.

6.6. CONCLUSIONS

In this chapter an architecture has been proposed for the development of ubiquitous web applications that interact with different types of data sources and involve semantic elements.

This architecture does not depend on a development technology on the server's side and includes a proposal for content generation process adapted to device groups.

The architecture considers one device grouping model that might be based on the mechanism proposed in this thesis or in any other way that might be developed in the future. This architecture serves as base for developments of multi-device web applications as a complement to existing proposals of logical modelling or as a base for a new modelling proposals. It also uses the proposals developed by the international community shaped in the recommendations of the W3C.

The implementation and testing stages of the framework were also described, as well as the special consideration that must be included in the development of ubiquitous web applications.

7. INTEGRATED COMPUTATIONAL ENVIRONMENT FOR EDITING UBIQUITOUS WEB APPLICARTION

7.1 SUMMARY

This chapter describes the internal elements and functionality of an integrated environment for the edition of ubiquitous Web applications that was created with the purpose of facilitating the incorporation of components that allow the treatment of device profiles within stylesheets that are edited by means of device independence principles.

Although there are editors incorporated within commercially available tools, these tools generate product dependencies, where they have been incorporated. Additionally, solutions that allow stylesheet editing within the framework of integration technology based on XML exist, but do not include profile treatment.

Due to this, in this chapter the internal components and the main forms of interaction through which a user can develop pervasive Web applications with this tool are described, which have been named IDEM based on the initials that make up its name (Integrated Development Environment for Mobile Devices).

Following, a description is given of how this environment was used in Linux servers based on Java Servlets, and how the Microsoft .NET environment tests were developed. With IDEM it is possible to generate scripts in C# that allow stylesheet treatment that incorporate the use of profiles based on UAProf.

Within the developed tests based on the .Net environment, the different configuration tests that were carried out with the purpose of generating independent device content are described, starting from the interaction with included classes within this environment for access to individual device capacities.

Finally a comparison of the use of these .Net and J2EE (Java 2 Enterprise Edition) environments for the development of ubiquitous applications is presented, based on the tests that were carried out trying to use IDEM for the edition of stylesheets based

on one of the languages available in .Net as well as Java Servlets. A comparative table of these two environments is also included.

7.2. INTRODUCTION

As a contribution to the above described problem, we have developed a tool called IDEM (Integrated Development Environment for Mobile Devices) that allows style sheets editing for content generation, as well as the generation of server-side scripts in different languages (Java Servlets and C # initially) with the necessary logic to transform to the appropriate language for different types of devices, considering the XML data source, the XSL stylesheet and the attributes of the devices (UAProf profiles of devices).

One first View of the developed tool can be seen in Figure 45.

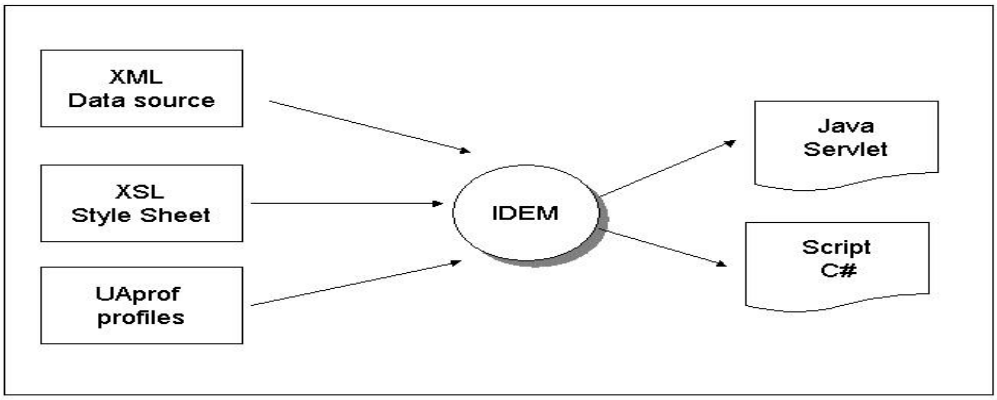
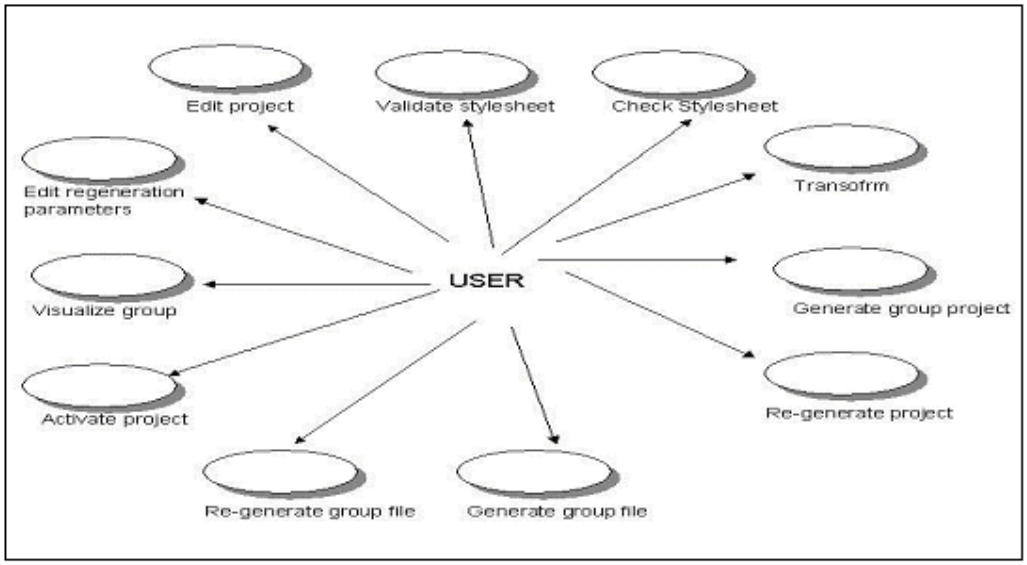


Figure 45. Extended View of the Operation of the architecture

7.3. INTERNAL STRUCTURE AND CHARACTERISTICS

The main elements that compose it describe in Figure 46, and the interface in Figure 47.



.

Figure 46. Use-Cases Diagram for IDEM

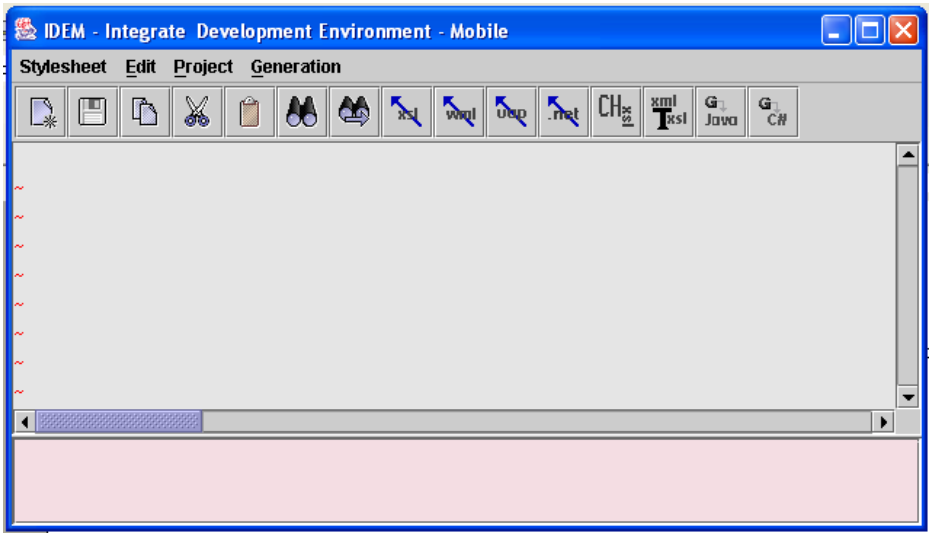


Figure 47. Interface of IDEM

IDEM offers diverse possibilities for stylesheet authoring, those related to the manipulation of archives, basic operations on the content, of syntax analysis, configuration, etc. Some of the included editing options are the following:

- Management of files: to create, save, remove and finalize.

- Operations on the content: to copy, paste, to cut, to search, and to replace.

- Operations on configuration: routes, options of configuration, etc.

Additionally IDEM offers functionality for the insertion of markup code such as: WML, CHTML, XHTML, XSL and HTML.

IDEM facilitates the stylesheets authoring verifying that they are lexically and grammatically correct. In case of a stylesheet that is not well formed, it reports the errors, indicating, the nature, the type of error, and, the line where it was detected. [See Figure 48]

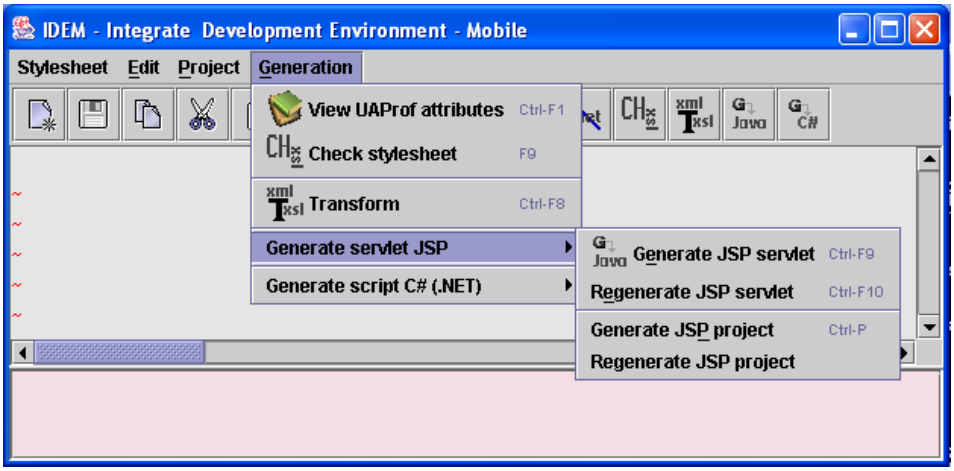


Figure 48. Verification of Style sheets in IDEM

Other characteristics of IDEM are the following:

- The specification of the parameters for the generation of the field with the logic for the transformation called template of servlets.

- The application of current style sheet transformation, applying it on an XML data source.

- It allow the generation of a standard template, in Java language or C#, the server-side scripts with the logic necessary to carry out the transformation of the source XML, applying the style sheet.
- Once the code of the script has been generated, the user according to specific necessities could modify the script. A basic aspect of consideration in the IDEM is the possibility to regenerate the code taking as the base the result of a previous generation, conserving any change made by the user.

The previously described functionality is based on the management of projects. A project consists of the relation of several specifications of generation, with the purpose of generating a template that includes, in the same script, different cases from transformation (according to the formatting language indicated in each file of generation).

In order to make agile the edition of the style sheets, it is needed to provide functionality that allows insertion of elements of XSL syntax, WML, HTML; as well as of attributes of the device profiles (UAProf).

The transformation functionality allows making formatting tests to the style sheet during the development process. The result of the transformation can be visualized in form of source code, or in some cases through the associated tool corresponding to the markup language generated by the transformation.

In order to carry out the transformation it is necessary to simulate the values of the attributes, of user and UAProf that are present in the style sheet. For this IDEM provides the necessary means to permit to the user specify these values when the transformation is made.

The scripts generated from IDEM have been tested on Linux based servers (See Figure 49), using Tomcat AS Servlet engine, and under Windows using Internet Information Server.

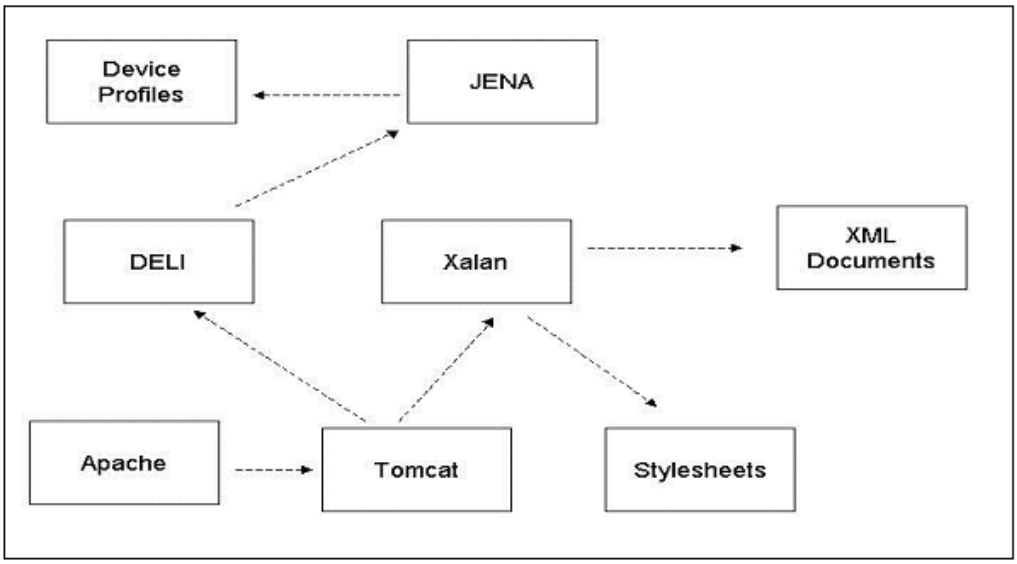


Figure 49. Processing of IDEM on a Linux environment

The integrated environment for development was implemented in Java, using four fundamental factors: support that offers Java for the object-oriented programming, platform independence (write once, run anywhere), the possibility of being able to reuse libraries developed by third parties, and perhaps the most important factor is its conception as language of development for the Web.

The more representative internal components of IDEM are shown in Figure 50 :

- JEdit a text editor developed in Java.
- Xalan, XML parser used for the style sheets transformations.
- DELI, opensource library for the processing of profiles.
- Transform, a Java API used with Xalan to transform stylesheets.

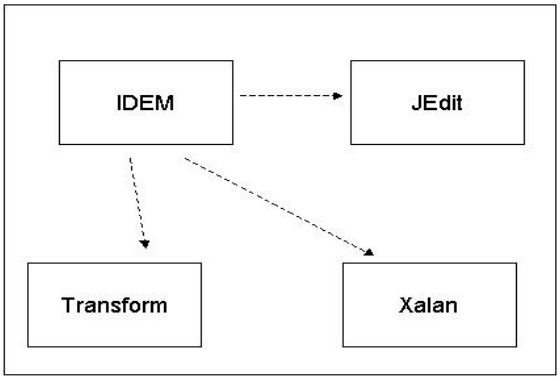


Figure 50. Main software components of IDEM

The conceptual frame is summarised in Figure 51.

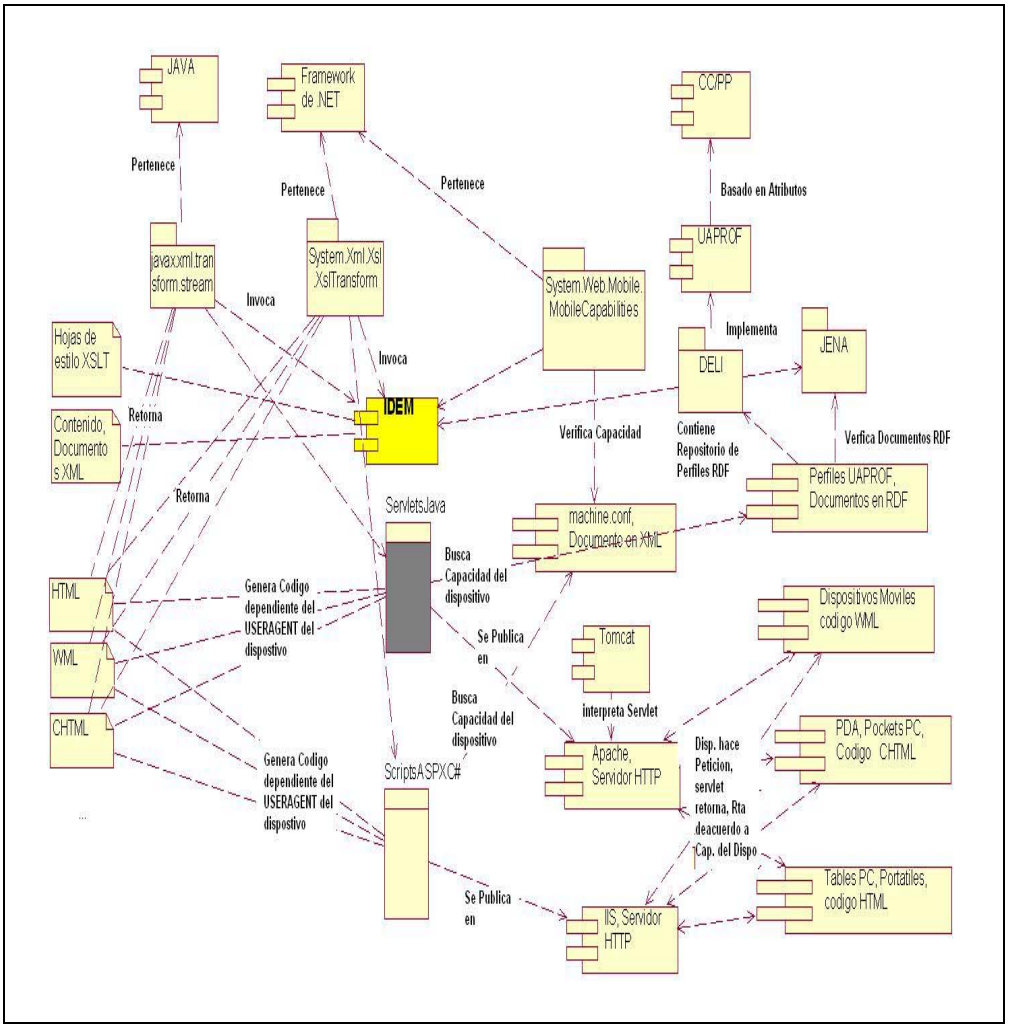


Figure 51. Conceptual frame of IDEM

7.4. EXPERIMENTS BASED ON THE USE OF DOT NET

The objective of this part was to generate scripts in C# from the IDEM managing the devices. Next, the understanding process has been detailed so as to meet the objective of generating scripts for .NET.

An important task was implemented when we study the main .Net concepts and making the devices independent. It is worth noting that the appropriation process is difficult due to the novelty of the .Net technology.

Generating code for .Net is the result of complementing IDEM with important tools so as to make it possible a comparison with that developed in Java and to verify the advantages and disadvantages that both technologies have.

As a first step, I studied the .Net technology in two introductory courses in mobile phones and C#.

The second step consisted in learning the installation, configuration and basic use of the .Net development environment, the Visual Studio framework.

The third step was the studying the language C#, which is a very difficult aspect since the C# is very similar to Java.

Investigating and learning .Net was carried out in the following way:

- The use of Servlets in .Net was checked taking into account the experienced made with the J2EE technology. [All02]
- The types and methods that allow interaction with the characteristics of the devices were also checked.
- The execution process of an ASPx was studied.
- The C# script template was specified, which similar to the one generated for Java.
- Implementation of the categories that generate C# templates.

- Integration of the functionality to the IDEM.

One of the points to be implemented when integrating IDEM with .Net was the query of the UAProf profiles from C# since in the investigation there is no reference of integration between .Net and UAProf.

Some tests in .Net were therefore made so as to interact with the UAProf profiles and the Java kinds of DELI. For that purpose, it was necessary to use Java classes from C#. There are tools that allow generation of libraries for .Net starting with the classes written in Java. One of them is JNBridgePro version 1.4.

The objective is to find DELI classes directly from a script written in C# or an aspx program. The library (DLL) is created as follows:

JNBridgePro is installed under Windows XP. (See Figure 52).

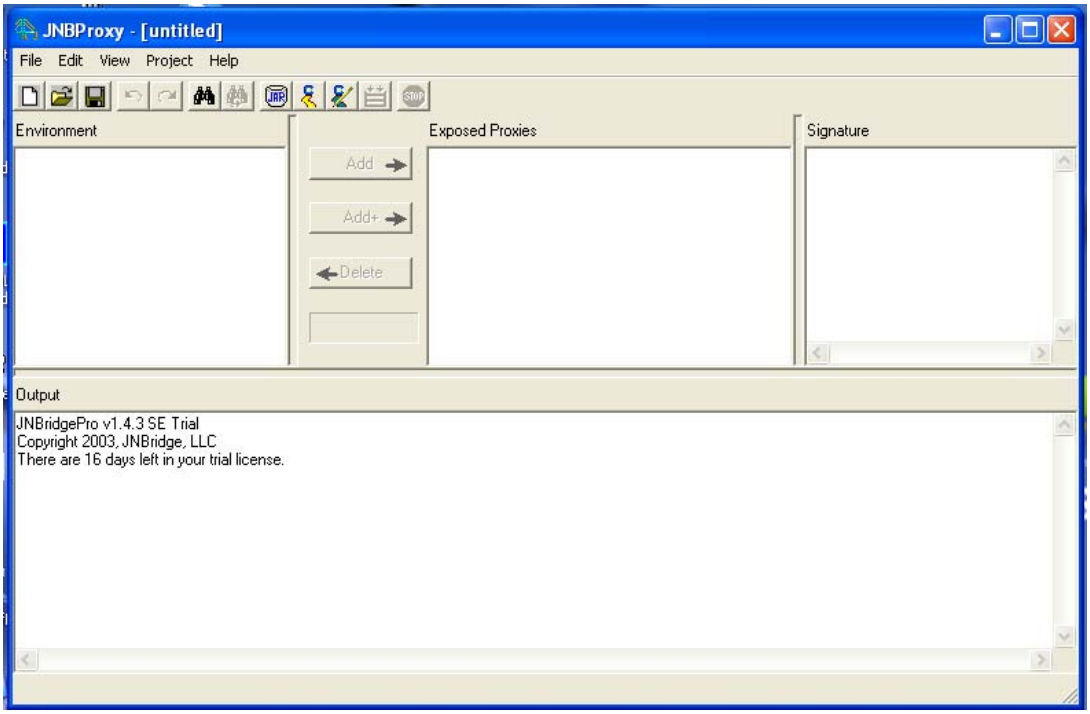


Figure 52. The IDE JNBProxy

The icon  Edit Classpath allows to define the directory where the DELI classes are found and the icon  Add Classes from jar file allows to add the packets to be transformed from DELI in such a way that the same methods can be called from .Net framework using C#.

After being selected, they will appear as (See Figure 53)

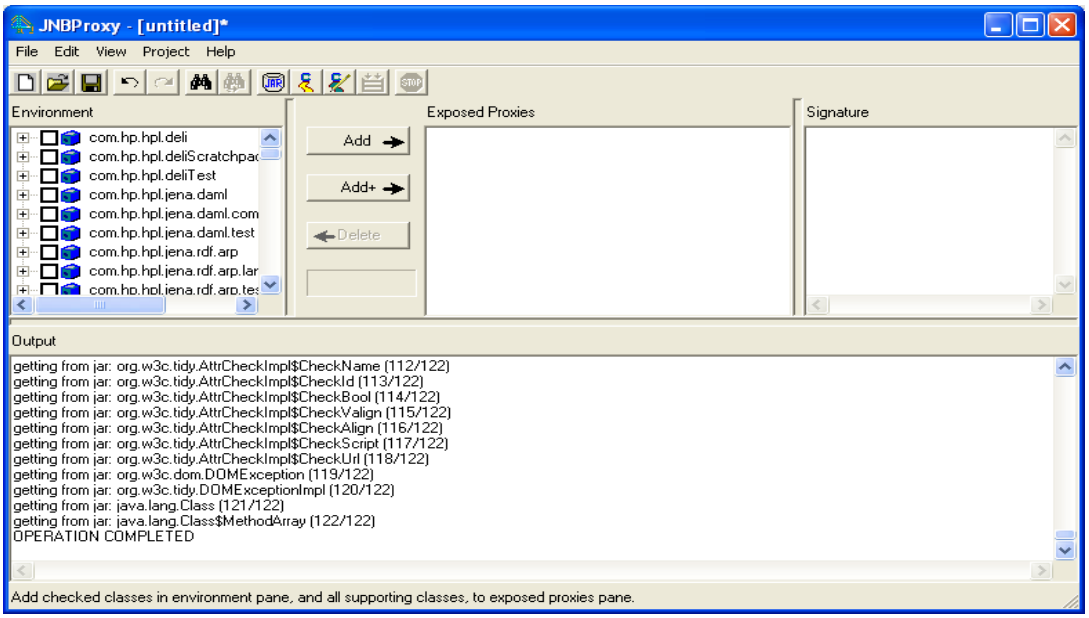


Figure 53. JNBPrtoxy – Java class selection

From the option “check all in Environment”, all classes of the DELI environment are selected as shown in the figure 54.

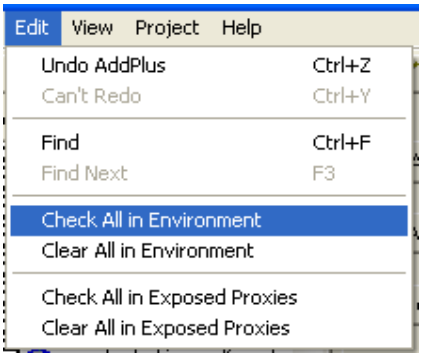


Figure 54. JNBProxy – all classes selected

The option Add+ is then selected so as to weight all the packets selected. If there are dependencies of Java packets or Tomcat packets, they should be added to the packet servlet.jar that belongs to Tomcat. The environment will be as follows: (See Figure 55)

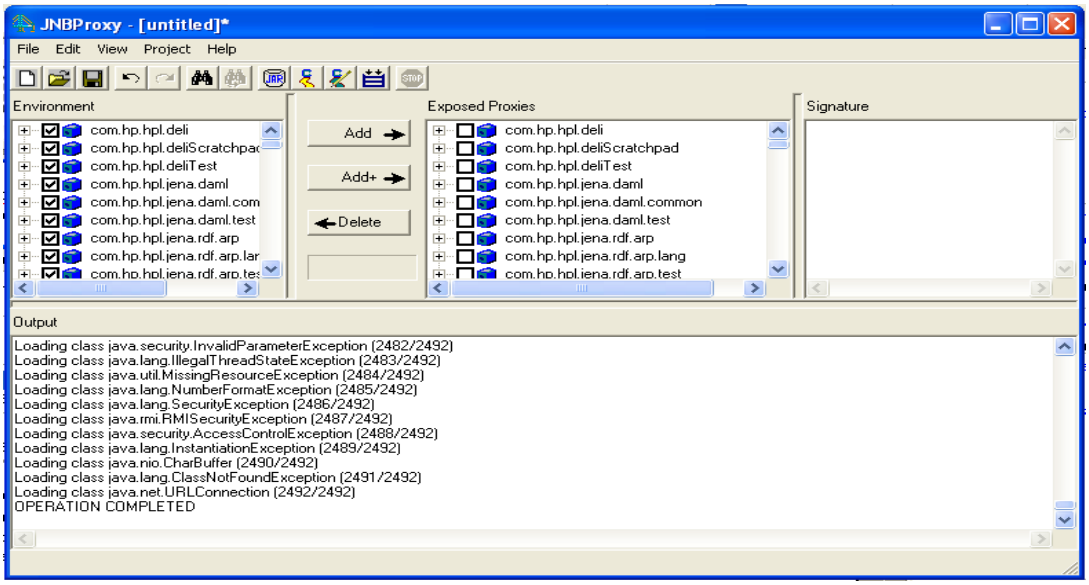


Figure 55. JNBProxy – DLL Generation

When there are no errors of dependencies classes, the DLL can be generated for the .Net Framework, the icon  (build) is selected, the name of the DLL is then requested as well as the directory where it is to be stored.

Once the DLL has been created, it can be called from a .Net application, referring to the path that contains the class that the C# class needs in this way:

```
using com.hp.hpl.deli;
```

The classes and attributes can then be accessed:

```
ProfileAttribute P;
string s = P.getAtrribute();
```

as it is done in a Java class.

Finally, the compiled .Net code is created referring to the library created with JNBridgePro.

```
csc /t:library /referente:Deli.dll Servlet.cs
```

Although the interaction of .Net Framework with the DLL was achieved, it was not possible to interact with passing the parameter from one environment to another. For example, the parameters that the Java ServletConfig receives are the same as those that the .Net class does. It was therefore necessary to load a large DLL that needed almost all the Java API. For that reason, the initial idea of integrating .Net and DELI was disregarded. Howver, it was considered, that it was a good alternative since DELI uses the Users Profiles more independently than in .Net.

By the previous thing it was necessary to load a so great DLL that it required of almost all the API of Java. For that reason we discarded this initial idea, to integrate .Net and DELI. Despite we considered that it is a good alternative, DELI handles the Profiles of the Users of an independent way but who what we found with Net.

The following steps was used to look for the way to handle the profiles of the devices in .Net.

The .Net Framework allows to develop applications that are independent of the context, interacting with XSL and XML and using the System.Xml.Xsl.XslTransform classes to load the profile sheet and the System.Xml.XPath.XpathDocument function to associate an XML document with the .Net Framework.

Additionally, the classes System.Xml.Xsl.XslTransform package used so as to make the call of the Transform method and make the transformation in the programming language necessary. In the programming of the code ASPx or script in C# , characteristics are defined that allow us to generate a code that is independent of the device.

.Net also has some classes that allow us to access the characteristics of a device through the packet System.Web.Mobile.MobileCapabilities, which inherits from the

class HTTPBrowserCapabilities. The class MobileCapabilities has an important number of read-only capabilities, which provide safe access to the dictionary of the objects capabilities in the browser.

The class Mobilecapabilities contains the standard characteristics of the capabilities of the ASP.Net browser. When a client is connected with a mobile ASP.Net application for Web, the page ASP.Net determines the request device and then it links the object of the class MobileCapabilities to the request.

When the page is being downloaded, the MobileCapabilities object has the capabilities through the browser characteristic. The PreferredRenderingMime and the ScreenCharactersWidth characteristics of the MobileCapabilities object return the answer to the browser fixing the object characteristics through the HTTPResponse. It is also possible to fix the characteristic of the device to a variable or Boolean chain so as to handle some controls in the page that are dependent on the capabilities of the device.

The following aspx scripts shows the use of MobileCapabilities.

```
<%@ Page Inherits="System.Web.UI.MobileControls.MobilePage"
  Language="c#" %>

<script language="c#" runat="server">

public void Page_Load(Object sender, EventArgs e)
{

  System.Web.Mobile.MobileCapabilities currentCapabilities;

  MobileCapabilities currentCapabilities =
    (MobileCapabilities)Request.Browser;

  // Programatically find the mobile capabilities without using
```

```
  // DeviceSpecific Filters.
  if(currentCapabilities.PreferredRenderingMIME=="text/html")
  {
    Label2.Text = "You are using an html supported device.";
  }
  else if(currentCapabilities.PreferredRenderingMIME=="text/vnd.wap.wml")
  {
    Label2.Text = "You are using a wml supported device.";
  }
  Label1.Text = "Screen Width (chars): " +
    currentCapabilities.ScreenCharactersWidth;
}
</script>
<Mobile:Form runat="server" id=frmTemplate >
      <mobile:label ID="Label1" runat="server" />
      <mobile:label ID="Label2" runat="server" />
</Mobile:Form>
```

The application can access the capabilities through the browser proprieties specifying an HTTPRequest object, which is mapped to the proprieties request of page request. The code of the application can access individual capabilities through a MobileCapabilities object in two ways:

First, it is possible to access any high-level characteristic. Each one of such proprieties, which is final, returns the value corresponding to the capacity of the device safely with its value associated. The example shows how a high-level propriety can be obtained.

```
if (((MobileCapabilities)Request.Browser).ScreenCharactersWidth > 20)
{   // Coding for big screen capabilities is placed here.}
else
{   // Coding for small screen capabilities is placed here.}
```

Second, The code can have access to the capabilities through the values of the capabilities dictionary. The values of the dictionary have been described in Machine.conf archive through the <browserCaps> section. This archive is written in XML. The returned values are defined as string data type, Boolean or null if the value has not been described within the group. It is responsibility for the application to capture and analyse the result returned. The following example shows the functionality of the dictionary.

```
String screenWidthText=Request.Browser["screenCharactersWidth"];

int screenWidth=(screenWidthText == null) ? 40 :
   Int32.Parse(screenWidthText);

if (screenWidth > 20)
{
   // This block contains code supporting a larger screen size.
}
else
{
   // This block contains code supporting a smaller screen size.
}
```

Next, the capabilities table of a device offered through the .Net Framework has been showed as well as its use in the different programming languages.

Once the previous tests have been finished, the template C# is then specified to be generated from the IDEM, to implement the IDEM classes for generation and to incorporate it in the IDEM functionality.

In conclusion, the integration of IDEM with .Net, a servlet is generated to be executed through the IIS and the .Net Framework environment with the characteristics that the Microsoft development environment presents.

7.5. DEVICE PROFILE IN DOT NET VS UAPROF

Capabilities of Nokia 7110 device seen from the archive machine.config in the .Net Framework and specified in XML format are the following:

```
</case>
        <case match="Nokia7110/1.0 \((?'versionString'.*)\)">
                type = "Nokia 7110"
                version = ${versionString}
    <filter with="${versionString}"
match="(?'browserMajorVersion'\w*)(?'browserMinorVersion'\.\w*).*">
                    majorVersion = ${browserMajorVersion}
                    minorVersion = ${browserMinorVersion}
    </filter>
                mobileDeviceModel = "7110"
                optimumPageWeight = "800"
                screenCharactersWidth="22"
                screenCharactersHeight="4"
                screenPixelsWidth="96"
                screenPixelsHeight="44"
</case>
```

The profile of the device Nokia 7110 defined by DELI within the Profiles directory and defined in language RDF
File: Nokia_7110.rdf:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:prf="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#">
  <rdf:Description rdf:ID="Nokia7110">
    <prf:component>
```

```
    <rdf:Description rdf:ID="HardwarePlatform">
     <rdf:type
rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#HardwarePlatform"/>
     <prf:BitsPerPixel>1</prf:BitsPerPixel>
     <prf:ImageCapable>Yes</prf:ImageCapable>
     <prf:InputCharSet>
      <rdf:Bag>
       <rdf:li>iso-8859-1</rdf:li>
       <rdf:li>utf-8; q=0.8</rdf:li>
       <rdf:li>iso-10646-ucs-2; q=0.6</rdf:li>
      </rdf:Bag>
     </prf:InputCharSet>
     <prf:Keyboard>PhoneKeypad</prf:Keyboard>
     <!--prf:NumberOfSoftKeys></prf:NumberOfSoftKeys-->
     <prf:OutputCharSet>
      <rdf:Bag>
       <rdf:li>iso-8859-1</rdf:li>
       <rdf:li>utf-8; q=0.8</rdf:li>
       <rdf:li>iso-10646-ucs-2; q=0.6</rdf:li>
      </rdf:Bag>
     </prf:OutputCharSet>
     <prf:PixelAspectRatio>1x1</prf:PixelAspectRatio>
     <prf:ScreenSize>96x44</prf:ScreenSize>
     <prf:ScreenSizeChar>12x4</prf:ScreenSizeChar>
     <prf:StandardFontProportional>Yes</prf:StandardFontProportional>
     <prf:Vendor>Nokia</prf:Vendor>
     <prf:Model>7110</prf:Model>
     <prf:TextInputCapable>Yes</prf:TextInputCapable>
    </rdf:Description>
   </prf:component>
   <prf:component>
    <rdf:Description rdf:ID="SoftwarePlatform">
```

```
      <rdf:type
rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#SoftwarePlatform"/>
      <prf:CcppAccept>
       <rdf:Bag>
        <rdf:li>application/vnd.wap.wmlc</rdf:li>
        <rdf:li>application/vnd.wap.wmlscriptc</rdf:li>
        <rdf:li>image/vnd.wap.wbmp</rdf:li>
        <rdf:li>application/vnd.wap.wtls-ca-certificate</rdf:li>
        <rdf:li>text/plain</rdf:li>
       </rdf:Bag>
      </prf:CcppAccept>
      <prf:CcppAccept-Charset>
       <rdf:Bag>
        <rdf:li>iso-8859-1</rdf:li>
        <rdf:li>utf-8; q=0.8</rdf:li>
        <rdf:li>iso-10646-ucs-2; q=0.6</rdf:li>
       </rdf:Bag>
      </prf:CcppAccept-Charset>
      <prf:JavaEnabled>No</prf:JavaEnabled>
      <!--prf:OSName></prf:OSName-->
      <prf:OSVendor>Nokia</prf:OSVendor>
      <!--prf:OSVersion>??.??</prf:OSVersion-->
      <prf:RecipientAppAgent>Nokia7110/1.0</prf:RecipientAppAgent>
      <prf:SoftwareNumber>1.0</prf:SoftwareNumber>
    </rdf:Description>
  </prf:component>
  <prf:component>
    <rdf:Description rdf:ID="NetworkCharacteristics">
      <rdf:type
rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#NetworkCharacteristics"/>
      <prf:SecuritySupport>
```

```
        <rdf:Bag>
          <rdf:li>WTLS-1</rdf:li>
          <rdf:li>WTLS-2</rdf:li>
        </rdf:Bag>
      </prf:SecuritySupport>
      <prf:SupportedBearers>
        <rdf:Bag>
          <rdf:li>SMS</rdf:li>
          <rdf:li>CSD</rdf:li>
          <rdf:li>GSM900</rdf:li>
          <rdf:li>GSM1800</rdf:li>
        </rdf:Bag>
      </prf:SupportedBearers>
    </rdf:Description>
  </prf:component>
  <prf:component>
    <rdf:Description rdf:ID="BrowserUA">
      <rdf:type
rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#BrowserUA"/>
      <prf:BrowserName>Nokia</prf:BrowserName>
      <prf:BrowserVersion>7110/1.0</prf:BrowserVersion>
      <prf:TablesCapable>No</prf:TablesCapable>
    </rdf:Description>
  </prf:component>
  <prf:component>
    <rdf:Description rdf:ID="WapCharacteristics">
      <rdf:type
rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#WapCharacteristics"/>
      <prf:WapVersion>1.1</prf:WapVersion>
      <prf:WmlDeckSize>1331</prf:WmlDeckSize>
      <prf:WmlScriptVersion>
```

```
        <rdf:Bag>
          <rdf:li>1.1</rdf:li>
        </rdf:Bag>
      </prf:WmlScriptVersion>
      <prf:WmlVersion>
        <rdf:Bag>
          <rdf:li>1.1</rdf:li>
        </rdf:Bag>
      </prf:WmlVersion>
    </rdf:Description>
  </prf:component>
  <prf:component>
    <rdf:Description rdf:ID="PushCharacteristics">
      <rdf:type
rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#PushCharacteristics"/>
    </rdf:Description>
  </prf:component>
 </rdf:Description>
</rdf:RDF>
```

7.6.COMPARASION OF THE ENVIRONMENT BASED ON DOT NET VS JAVA

There are debates as to the relative merits of J2EE versus .Net strength. Some observers feel that Microsoft has the lead beacuse of its strong support for XML standards, such SOAP; others feel J2EE will be dominate beacuse of the strong industry support for this platform from other software leaders.

The advances being made and the trajectory that Java has in the programming world have a lead over the .Net world. However, in the near future, the number of APIs and the environments developed for the .Net Framework may be equal in number or even surpass the ones in Java.

The IDEM tries to present to the user some alternatives for publishing pages that are independent of the device for both development environments. However, the Java environment currently has advantages over the .Net environment. Next, the advantages and disadvantages of using Java and .Net in a development environment called IDEM have been specified.

One advantage is the number of APIs and development environments that are found in Java, that enable work with platform independency, whereas in .Net the environment being used is integrated are new, with an propieatary API for the request and verification if devices capabilities.

The HP's DELI API enable to capture the characteristics of a device profile. A new device profile is published by the vendor and it is defined in RDF. When the request for an attribute that has the capability of a device is handled from programming, it is loaded through the class DELI Profile and PerfilAttribute, which the capability to verify is passed as a parameter. Through this the environment is verified and the value is obtained.

This procedure allows the loading of the capabilities of a device and the verification is carried out dynamically, which is defined every time that a new capability in a device is created or when the value of the independent capability of the development environment changes. All the capabilities are captured through RDF resources archives and, if a new version of a device is created, it is only necessary to create an RDF archive for the device to integration. Through DELI, the capability can be requested.

The .Net Framework specifies some device capabilities through the class System.Web.Mobile.MobileCapabilities, which allow us to know whether or not the device has a capability that is wished. The capabilities that can be referenced depend on the ones that the Framework specified in that class. It is not possible to add more, since there is no access to the source code so as to make any modifications.

When there is a request for the value that the lost capability has, the .Net Framework obtains such value from a configuration archive called machine.conf, which is being

written in XML. The values described are linked to the capabilites that the class has, through which the capability value is accessed. If the value of the capability of the device changes, it is possible to go to the configuration archive and change the value. However, if the device capabilities are the ones that change and such capability is not specified in the class attributes, it will not be obtained.

The capabilities that the .NET framework defines do not follow the CC/PP standard the structure for the XML archive, in the name or in the values of the attributes. That is, the .Net platform specifies its own standard.

The scripts and the pages that have been developed with Java may be published for almost any hardware platform and under any operating system thanks to the distributions of the Java virtual machine. The lending content through servlets with Tomcat may also interact with almost any HTTP server including Microsoft ISS.

Due to the closed and proprietary nature of Microsoft, it is very difficult to interact with a server that is not ISS from Microsoft and with operative systems with architectures that are different from the architectures from Microsoft.

There are no recommendations from the international community or the W3C consortium, which are based on the separation of content and presentation, which may be observed in the adoption of proprietary mechanisms for format specification, different from CSS and XSL that had been adopted by the community.

7.7. CONCLUSIONS

In this chapter the internal structure and functionality of IDEM were described. IDEM is an environment for editing and developing of ubiquitous portals, based on the facility of the authoring of stylesheets with support of processing of device profiles, as well as on the generation of server-side scripts that facilitate the transformation to different types of formats by means of the interaction with stylesheets and XML documents. The raised solution does not generate any vendor dependency and can be used in authoring processes of content for different server platforms.

Due to the closed .Net a proprietary solution, it is very difficult to interact with a server that is not ISS from Microsoft and with operating systems with architectures that are different from the architectures from Microsoft.

The .Net Mobile Internet Toolkit doesn't adhere to the recommendations as expressed by the W3C consortium which are based on the separation of content and presentation. A clear example is the adoption of proprietary mechanisms for format specification, different from the standards commonly adopted by the community: CSS and XSL.

It is not necessary to say that the open-source developments, defined by DELI make strict use of the W3C's recommendations for generating content and presentation.

SECTION III. EXPERIMENTS AND VALIDATION

8. EXPERIMENTS DEVELOPED DURING THE RESEARCH PROCESS

8.1. SUMMARY

This chapter presents the description of different prototypes and applications for Mobile Internet that were developed at the same time as the theoretical revision process, as well as the initial tests based on device independence solutions.

In this outline, the functionality and internal architecture of WAP applications for information management of traffic incidences that were developed for SCT (Servei Catalá de Transit) and DGT (Dirección General de Tráfico) are included, which did not include treatment of differences on device capabilities nor device independence principles, due to the fact that at the time of this development there was not complete understanding of the W3C's standards and proposals.

Additionally, the description of a Web application prototype based on voice navigation is included, which allows interaction with meteorological data found in a database. The experience obtained in the development of this prototype served as basis for the design and implementation of a traffic incidence portal for the SCT based on interaction by means of conventional telephones.

Additionally, an analysis of how the design of these applications could be improved if the framework proposed in this thesis had been used is included

Furthermore, the tests developed with the DELI library and with Microsoft Mobile Internet Toolkit for generation of heterogeneous content accessible from multiple devices is described, which served as a practical basis for the applications based on the framework that is described in this thesis.

The last prototype that is described in this chapter is based on the construction of a search interface that can be accessed from WAP devices as well as I-Mode, and interacts with the DELI library for the treatment of device capacities. Text-summarization techniques for the interaction with a search engine based on HTML were used in this prototype, later translating the markup languages.

8.2. INTRODUCTION

In this chapter, the prototypes and mobile applications that were developed in the initial stage of the investigation process are described, when the main proposal developed in this thesis had not been specified (sections 8.1 to 8.6).

The development of these tests has been of great importance in obtaining a full understanding of the diverse technologies related to the development of ubiquitous applications.

At the end of each presented case, an analysis is made of the way in which each application or prototype can be improved if the framework proposed in this thesis had been employed.

Likewise, tests and prototypes are described, based on the use of device independence principles. These tests included the configuration, installation and testing of the DELI library as well as the Microsoft MIT solution. Additionally, the characteristics of a prototype that was used to verify the way of using the DELI library in a content generation project for multiple devices are described (sections 8.7 to 8.9).

8.3. DEVELOPMENT OF A WAP PORTAL FOR TRAFFIC INFORMATION FOR THE SERVEI CATALÁ DE TRANSIT

Within the different activities of this project, technologies were studied that allowed Internet access from mobile phones such as the WAP protocol: The project resulted in the first wireless solution for traffic incident information on highways of Catalonia, Spain.

For this project it was decided to use Oracle as a DBMS, instead of the existing legacy database; as a system for the storage of information related to incidents. The additional advantage of this was to be able to offer the greater scalability needed to allow for simultaneous access by a high number of users.

In this phase of the tests we didn't provide a separation between content and presentation, since there was not enough understanding about the link between XML and the version of the database available at the moment. Content management generation was done using PL-SQL scripts.

In this phase of the proposals device independence did not exist. Nevertheless, we obtained a good degree of understanding the WAP protocol and the associated markup language.

A simplified structure of the dynamic interaction that is presented in this application can be seen in the Figure 56.

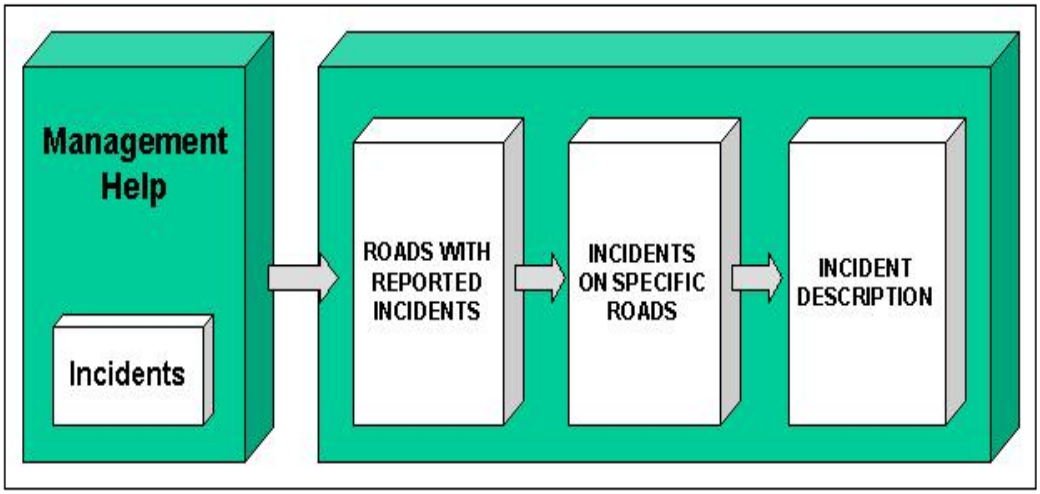


Figure 56. Dynamic Interaction in SCT's WAP portal

In the main menu, static information is offered about different matters related to fines, addresses and relevant phone numbers, traffic constraints, safety and traffic education.

The "*Incidences"* option allows the user to obtain information about reported incidents inside the SCT (Servei Catalá de Transit, Catalonian Traffic Administration) database in real time. Furthermore, it is possible to filter the incidents, except for those caused by road works that are recurrent, about which drivers are aware.

Access to this information is by choosing the following options: Incidents in all Catalonia, by provinces, regions or road (see Figure 57).

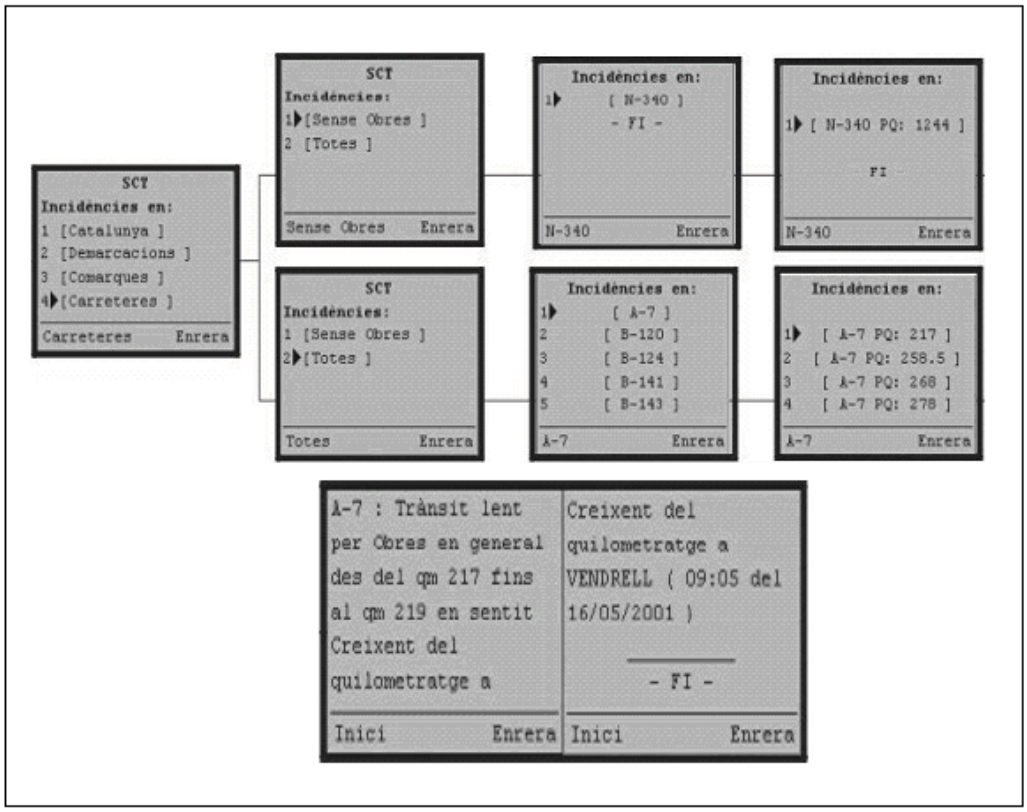


Figure 57. WAP interface for SCT project

Navigation was simplified as far as the number of clicks is concerned and this meant a reduction on the temporal and economic costs. For each selected incident it was possible to display an incident description with details about the causes, duration and location.

If this application were to be redesigned employing the framework proposed in the thesis, the application maintenance process and the scalability would be improved, since an efficient way of giving support to new devices would be available. Also, the maintenance process would be eased, since file updates would only be required for those that define the visualization format, but not for all the application files.

8.4. DESIGN AND IMPLEMENTATION OF A TOOL FOR AUTHORING OF DYNAMIC WAP PAGES

In this project a WML compiler was implemented, which enabled the generation of dynamic content from different server-side technologies such as ASP, PHP and JSP.

The tool facilitates graphical authoring of this type of documents and allows a syntactic verification of scripts. Device independency was not included in this project.

In the development, we identified the following requirements:

- The main functionality of the software to develop is to assist to the generation of the user code.
- It must assist to the generation of the WML code (versions 1.1, 1.2 y 1.3), WMLScript, ASP, PHP, and JSP.
- The functionalities show to be similar to the typical text editor.
- It must to be possible to edit several files and languages simultaneously.
- It is necessary for a WML code analyzer to indicate visually the errors on the code and make it easy to correct them.
- There must exist the possibility of launching a simulator to show the results.

Furthermore, the following two requirements are important too:

- Translate WML code to the WML 2.0
- Assist in the generation of this new language.

The construction of a grammar set and finite automata, which give support to the development of an editor/compiler of dynamic WAP pages, is presented.

The compiler provided support for WML and WMLScript, as well as the possibility of using different technologies for interaction with databases on the server side, such as ASP (Active Server Pages), JSP (Java Server Pages), and PHP (Hypertext Processor), allowing the utilization of the defined elements in the WML 2.0 specification.

A lexical, syntactic and semantic WML code analizer was designed, this allows for error detection in the authoring with a detailed description, which is important in the revision of documents developed manually as well as in the revision of documents previously created.

Additionally, a translator was developed that allows the transformation of written pages in older versions of WML and WML 2.0. The translator is useful in the creation of a grammar and finite automata with which to analyse the code.

The final software interface facilitated the editing of WML code and the inclusion of the tags presented in the technologies for the integration of Web applications and databases: ASP, JSP and PHP. The software prototype has the ability to simultaneously edit multiple files.

For the prototype development, object orientation methodologies were used, and the design was specified in UML (Unified Modelling Language) using C++ as the programming language.

A syntactic and semantic lexical analyzer was designed with WML code, which allowed for error detection during editing and a detailed description of same, which is of importance for the revision of new or existing documents. Additionally, a translator was created, which allows for the transformation of pages written in earlier versions of WML into the newer WAP 2.0 format.

The micro-browsers found in mobile terminals have an interpreter for WML and one for WMLScript. The latter, also implements a compilation of libraries which permit the scripts to access certain services at the terminal.

Moreover, it has the additional function of telephone communications by means of a WTAI (Wireless Telephony Application Interface) agent, through which it is possible to obtain access via the scripts to specific functions in order to realize tasks such as: call initiation and access to telephone directories.

The final software interface, which was developed facilitated the editing of WML code and the inclusion of the tags presented in the technologies for the integration of Web applications and databases: ASP (Active Server Pages), JSP (Java Server Pages) and PHP (Hypertext Processor). It was added to the authoring tool the ability to simultaneously edit multiple files.

The final developed prototype allows the possibility for supporting technology in the development of dynamic scripts on the server side which interacts with databases by means of technologies such as ASP, PHP, JSP, and, additionally, of providing support for the conversion of documents to WML 2.0 and the facilitation of error detection.

Finally it is important to emphasize that the development of this complier served as the basis for the construction of the integrated environment development proposed in chapter 7 of this thesis. The main useful elements were the syntactic and grammatical analysers. Regardless of the fact that the files generated with this compiler included presentation formats embedded within the server side scripts and that the stylesheet supports were not included, the experience obtained was important in conceiving the environment.

8.5. DEVELOPMENT OF A WAP SOLUTION FOR ROUTE GUIDANCE COMMISIONED BY THE DIRECCIÓN GENERAL DE TRAFICO

The objective of this project was to develop a system that allows the display of route information on WAP mobile devices with an efficient and user-friendly interface. It also generates information related to traffic incidents associated to the selected route.

This portal provides the possibility for users to plan a route by highway, associated to different criteria such as the possibility of choosing the fastest or the shortest route.
This specific development was necessary due to the need to generate WML content from a Windows server using Active Server Pages. This specific platform was selected because of the existence of a series of libraries for this platform. In this way a new platform for server generation of dynamic content was tested.

Finally, if the proposed framework were used in the development of this prototype, one of the major advantages obtained would be the independence of the platform and the better organization of the application files. Additionally, the itinerary generation library should be modified (which was developed by a third party) in order to generate XML outputs instead of HTML, with which it would be possible to include posterior transformation processes.

8.6. VOICE PORTAL PROTOTYPE OF METEOROLOGICAL INFORMATION

This was prototype was based on a database request through voice commands. The trials were developed with an Apache server [Apa01] with PHP, technology for the development of dynamic applications, and with MySQL, a database server.

The prototype is available with a telephone connection or with a PC connected to the Internet that has a browser with voice support.

It presents weather information of some European cities, based on the following data:

City	Temp.	Hum.	Visibility	Wind
Paris	55	53	Unlimited	From the west at 14 mph
Rome	75	84	Unlimited	From the Souteast at 7 mph
London	52	91	Unlimited	From the south at 5 mph
Madrid	75	84	Unlimited	From the southeast at 7 mph

Table 7. Data stored in the meteorological database

Note: Temp colum represents temperature in degrees Fahrenheit and Hum represents humidity in percentage.

When accessing the Internet from a telephone, the client makes use of the VoiceXML gateway of Voxpilot in a transparent way.

The gateway carries out the recognition and voice synthesis tasks for free as long as it is not for commercial purposes. The only fee charged is when calling the number to test the service developed.

In order to access the application the user dials the local phone number in Spain **917 889 537**, which is offered by the free-gateway VoxPilot.

When initiating the dialogue, the application asks the user the Caller-ID, which is assigned by the host service, in this case it corresponds to 61 21 21 21. Later the PIN will be asked, in this case it is 1234.

Once the process of user identification has been completed, it will be presented with a welcome message and a list of cities, the user will then name the city of interest and the relevant information will be given.

A graphical scheme of the operation can be seen in the Figure 58.

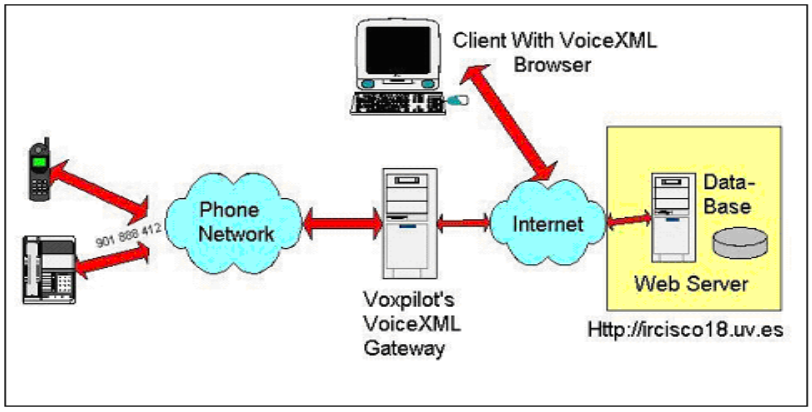


Figure 58. Architecture of Meteorological Voice Portal Prototype

The application can also tested from a computer connected to the Internet, as illustrated in the Figure 59.

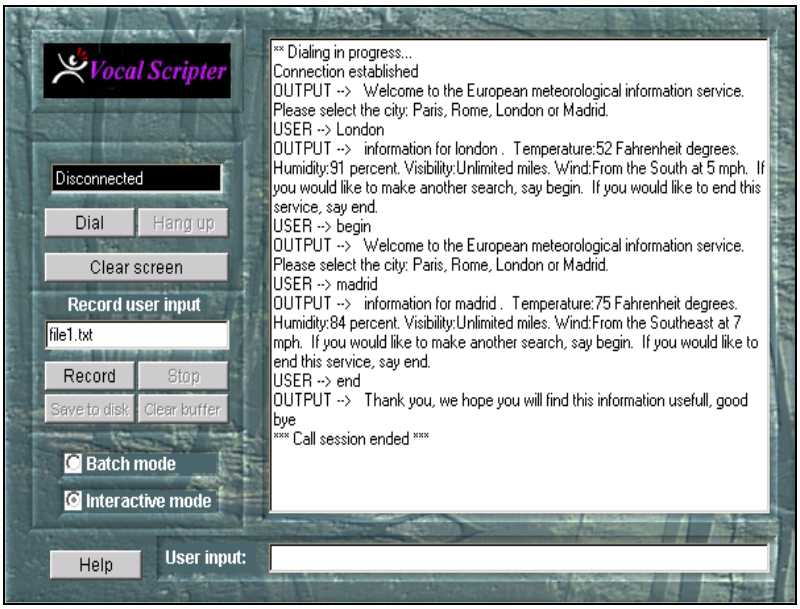


Figure 59. Text based Interaction

If this application had been developed using the proposed framework, better information organization could have been achieve, mainly due to the use of stylesheets, which constitute a better alternative than the approach based on the use of Voicexml embedded in the scripts.

However, the group processing included in the framework would not be necessary in this case, due to the application being developed solely for access by voice.

8.7. IMPLEMENTATION OF A VOICE PORTAL FOR THE SERVEI CATALÁ DE TRANSIT

This project was initially conceived as a development to use new technologies to enable voice navigation of Web content.

In this project initially, a study was made of the state-of-the-art of accessing the Web through voice control. The different elements were studied that compose a system of synthesis and voice recognition for the Web. It also determined the functions of the different elements involved in the process, and cases and applications of this technology.

Besides that, several functional prototypes were developed using different technologies on the serverside using mainly PHP, JSP and Oracle

The internal architecture of the system is shown in Figure 60, and a typical dialog used in the system is shown in Figure 61.

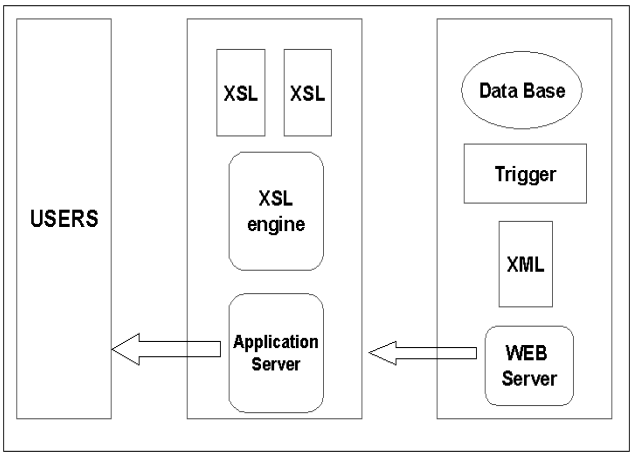


Figure 60. Internal architecture of the SCT's Voice Portal

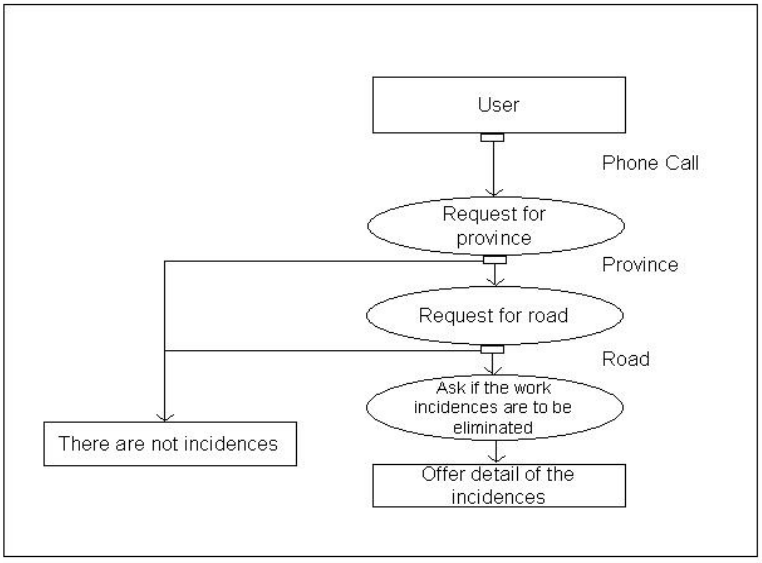


Figure 61. Dialog uses in the SCT's Voice Portal

Finally, this application is prepared to be used parting from the framework, since it is based on the development of stylesheets and XSLT transformations and not on the simple generation of voicemxl markup. However, multi-device capabilities were not

included because only voice interaction was considered in the conception of the project.

8.8. CONTENT GENERATION FOR PDAS

These tests were based on the study of the mechanisms of accessing Web contents using IPAQ PDAs.

The tests were initially based on emulated devices. First the Pocket PC emulator from Microsoft Windows Platform SDK for the PocketPC was used.

In a later stage of the project, the functionality was implemented in real devices. It was based on synchronization of the connection with an Internet connected PC, and from PDAs connected to the network using GPRS or bluetooth.

The Figure 62 shows the interface that was visualized using the emulator.

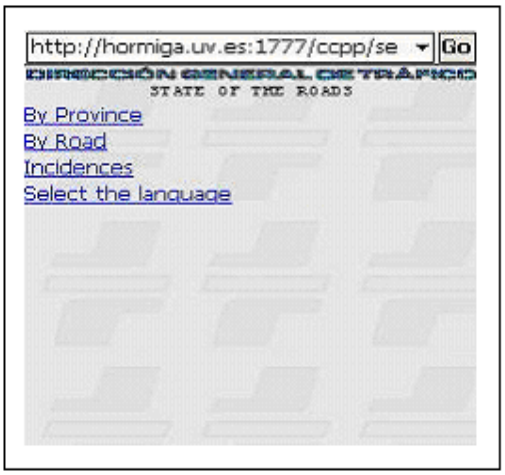


Figure 62. PDA interface using the Microsoft's emulator

The importance of these tests in relation to the proposed framework is based on the possibility of verifying the correct generation of content, adapted to different PDA models starting from the analysis of their capability.

Also these tests have help in obtaining an improved clarity in the navigational and hypertextual structure differences between different devices. In this sense, it was observed how a list box with all the Spanish provinces could be included in a PDA, meanwhile in WAP devices a subdivision of two or more screens had to be made.

8.9. CONFIGURATION, INSTALLATION AND TEST OF DELI LIBRARY

By means of these tests it was possible to get understanding of the operation of the DELI library and its capacity to handle device profiles from UAProf and CCPP.

The results were verified with emulators and with real devices such and with the browsers Internet Explorer, Netscape and Openwave emulator. (See Figure 63)

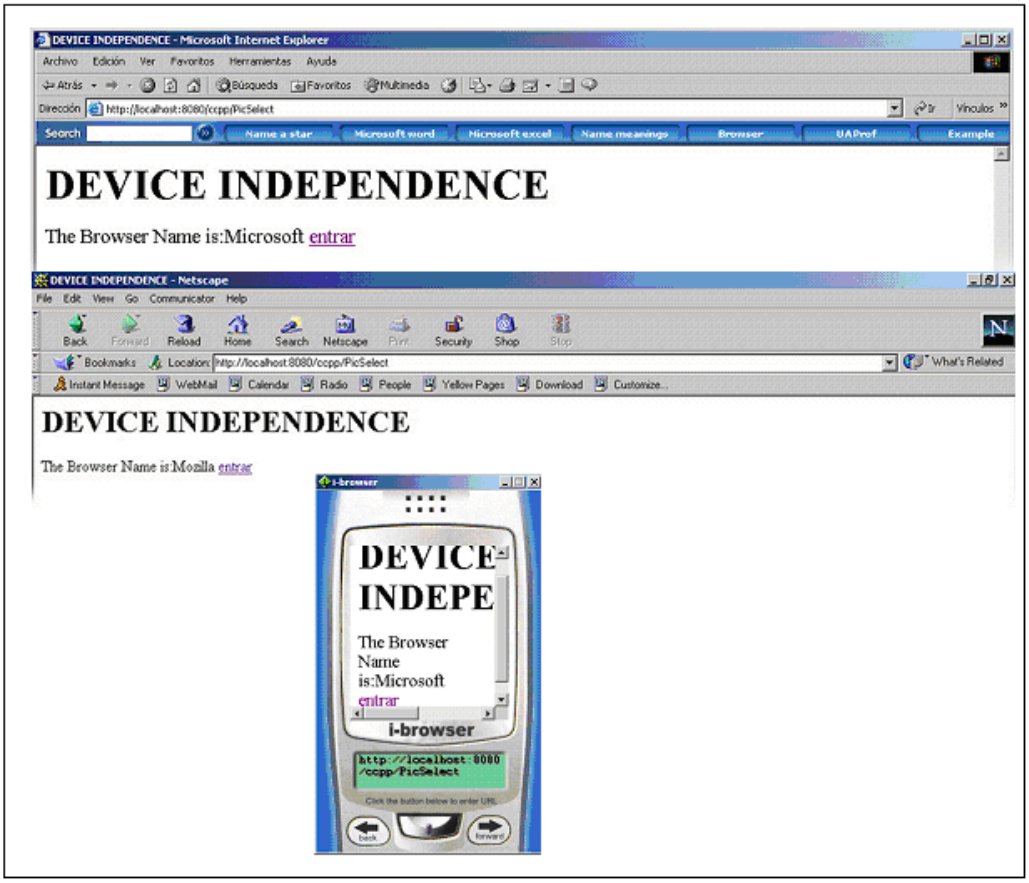


Figure 63. Test of DELI library in different browsers.

8.10. EXPERIMENTS WITH MOBILE INTERNET TOOLKIT

In relation to the tests developed by the Microsoft's Mobile Internet Toolkit-ASP Mobile Controls (MIT) proposal, the process that was implemented is as follows:

Having a . Net platform installed, a graphic interface was created based on a Mobile Web Form composed by fixed text (labels) text boxes and a command button.

These controls were added to the form using the drag and drop technique. Then the last update of file machine.config was downloaded, which contains devices recognized by MIT.

Finally, the application was executed and the correct interface visualization was verified, in emulators as well as in real devices. This was obtained since. Net is in charge of carrying out a compilation and transformation process, in which the automatic conversion of language markup is supported by the device formulating the query.

Mobile Internet Toolkit (or ASP.Net mobile controls as it now called) targets mobile devices and renders the correct markup, which is then sent to the requesting device. All this happens on the server.

However, the following disadvantages of this approach could be observed:

- If specific profile aspects of the device are to be employed, the program language selected for the interface development (i.e. C#) must be used. It is not possible to use stylesheets for the profile process, even though a component called stylesheet exists, as this does not really allow use of the stylesheets in standard form proposed by the W3C.

- It was not possible to use the graphic generation process of the interface in an environment that is not Windows, since .Net was only available for this kind of servers.

8.11. DESIGN AND IMPLEMENTATION OF AN INTERFACE FOR INFORMATION RETRIEVAL WITH MOBILE DEVICES

This project was focussed on the information retrieval from mobile devices.

Traditional search engines are not suited for mobile devices, many search engines are still desktop oriented and not well suited for mobile devices, because:

- They cannot be accessed with a mobile device, because the search engines do not have a user interface for the markup language the mobile device uses.

- They find content that cannot be shown on the mobile device.

- The search query page or the result listing does not fit on the mobile device's screen or in its memory.

- They lack regional information, with mobile devices often want to restrict the query or the location they currently are in (for example, find restaurant nearby).

As a way to deal with this problem, a search engine for mobile devices prototype was developed. For queries, a third party metacrawler HTML search engine was used. Currently the search engine support, WAP and I-Mode phones.

The developed work consisted of the elaboration of an interface that allowed performing a search, by calling a HTML search engine and converting the results from HTML to WML, using text summarization and filtering techniques. The solutions also used the DELI library to solve the profile.

Additionally there was an objective to study the I-Mode protocol and to check the functionality of the portal with I-Mode emulators. The final application was tested from WAP and I-Mode emulators [Pix01,X901], as well as with real WAP mobile phones.

The I-Mode interface based on the J-Phone Emulator is shown in Figure 64.

Figure 64. I-Mode Interface for the search engine prototype

The WAP interface for the Search Engine is shown in Figure 65.

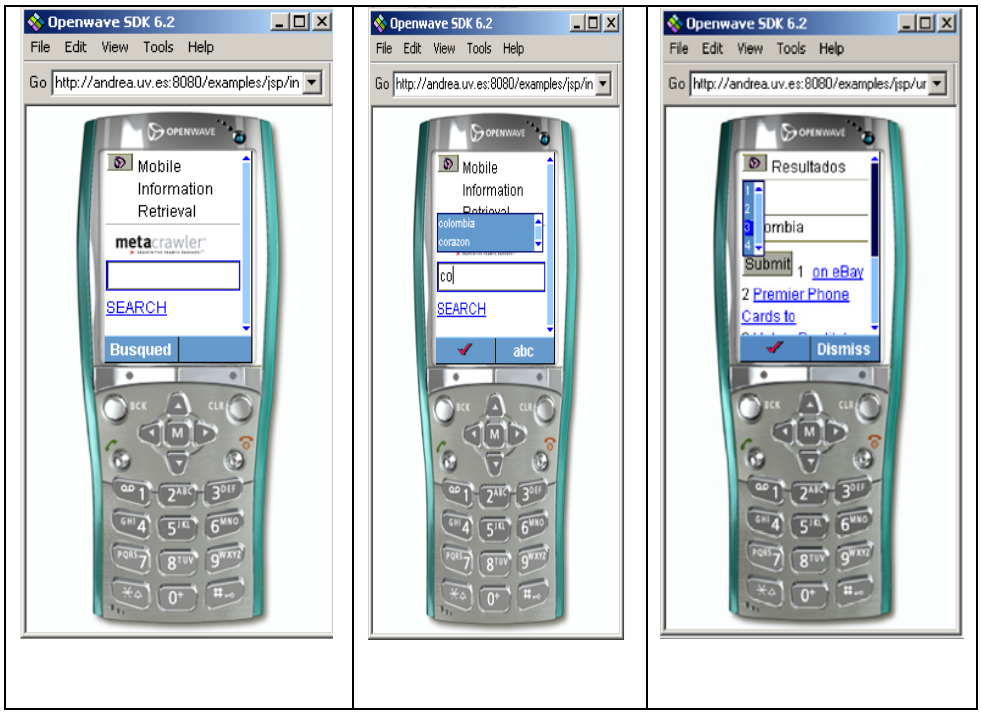


Figure 65. WAP Interface for the search engine prototype

The web interface based on XHTML is shown in Figure 66

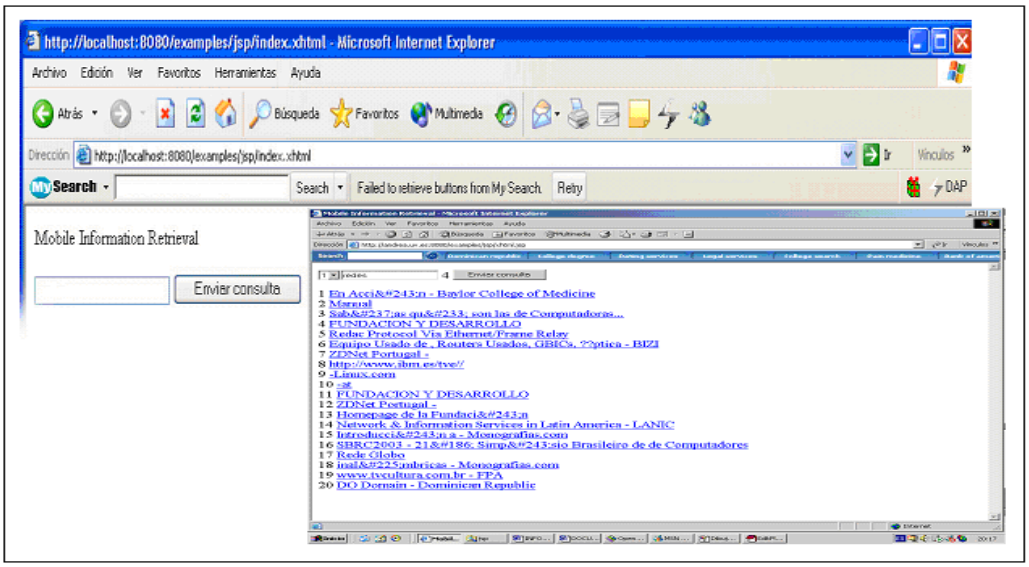


Figure 66. Web Interface for the search engine Prototype

For content in HTML, no simple way exists to transform it into other formats, even if it follows the HTML 4.0 specification. There are too many parameters that can be interpreted differently. Indeed, a high proportion of the available documents are

incorrect. In summary HTML content is not easily usable in transformation that enables control over content in the manner that an author should expect.

Finally, the result of the searcher are displayed in a mobile phone using text-summarization techniques (See Figure 67)

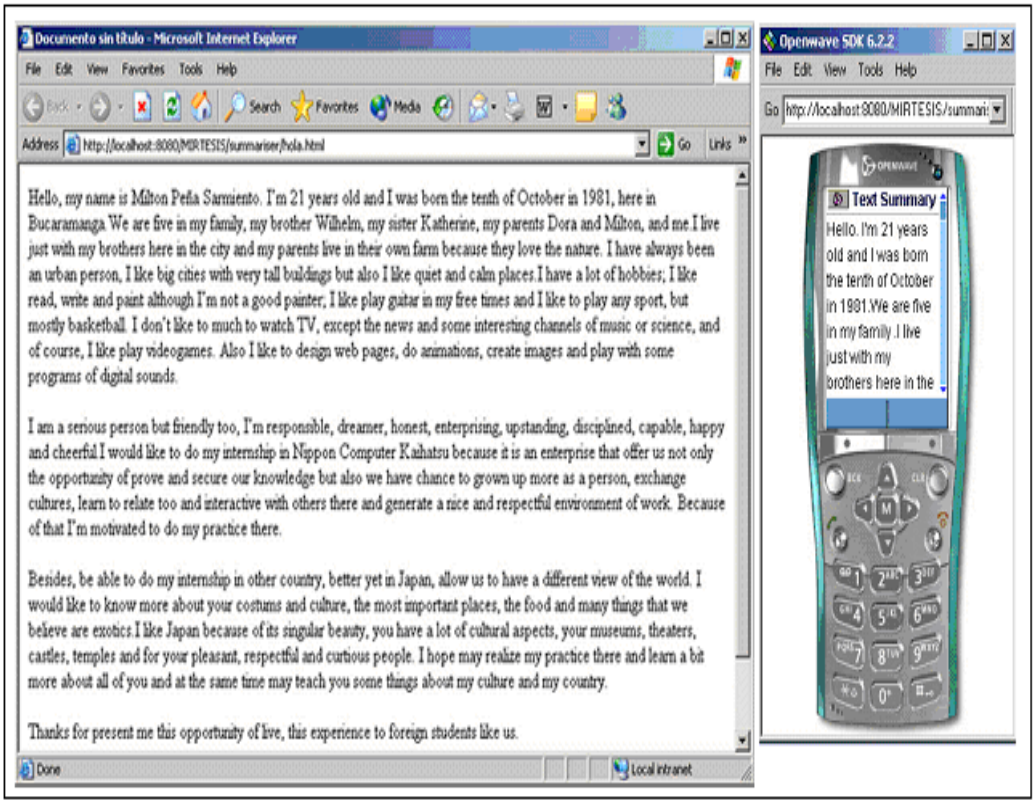


Figure 67. Extraction based on text-summarization techniques

8.12. CONCLUSIONS

Multi-device web applications, and in general the access to the Internet content, is useful for Intelligent Transport Systems since the Internet is the only worldwide and wireless media that enables people to get real-time and personalised information

Mobile phones are already being used by millions of people in the world, far more than car navigation systems can hope to achieve in years. A wireless Internet mobile phone, could therefore be seen as an alternative to traditional car navigation systems.

A micro-navigation device on the mobile terminal can receive a limited portion of data corresponding to the route computed and its immediate surrounding.

However, it is not safe for a driver to be reading guidance information on a tiny screen. It would be even more dangerous than having a conversation on a mobile phone and holding the steering wheel with the other hand. Therefore, it would be better to use wireless access to the Internet with mobile devices before starting a trip. An Internet access phone also allows pedestrians to ask for traffic conditions or a transport company to plan the trips of its vehicles.

In relation to the use of voice applications in Traffic Management Intelligent Systems, the following has been concluded:

- The access to any type of information while driving will be much safer with the use of voice applications or for those vehicles that have hands-free access. A service of voice alerts and a voice application on traffic incidents allow information to be accessed about the road conditions in real time, while at the same time it is not a danger for the road users.

  In the case of the voice application, it is possible to include a wide range of information apart from the traffic incidents, such as important telephone numbers, trip advice or the nearest gas stations and restaurants. With this information, the user can plan his/her journey itinerary while driving and can avoid any potentially dangerous situation, making a good distribution of the traffic density possible.

9. IMPLEMENTATION OF A UBIQUITOUS WEB APPLICATION FOR TRAFFIC INFORMATION USING THE PROPOSED FRAMEWORK

9.1. SUMMARY

This chapter describes the process followed in the development of the main pervasive Web application that has been developed, based on the framework proposed in this thesis. The developed application is based on the interaction by means of WAP and I-Mode telephones, PDAs, and conventional telephones, with traffic incidence information, which is available in XML format or from a URL.

Initially, the main requirements that were indicated by DGT as well as the benefit that can be obtained by the users starting from the interaction with this application are described.

Afterwards, the main characteristics and limitations that existed in the first initial version available for this application are listed. This is the version which should be redesigned. In this first application not even the separation of content and presentation existed, since the content generation for WAP devices based on WML language was absorbed within existing scripts alongside the server.

Additionally, a first approximation that was developed in search of an efficient separation of data and format is described, which was based on the use of XLSLT transformation techniques and the inclusion of stylesheets for different types of devices.

In the presented description, diagrams that are generated starting from the modelling of conventional techniques are included, as well as navigation diagrams that summarize the system interaction. Additionally, an instance of proposed architecture in the framework for the physical modelling of this application, is described. Finally, part of the interface that is visualized in PDAs as well as in based WAP telephones is presented, which allow graphic interface elements such as in conventional WAP telephones from previous models.

9.2. INTRODUCTION

The most important aspects related to the implementation of a ubiquitous web application are described in this chapter. The development was based on the proposals raised in this thesis and served to validate and to clarify the proposal.

The work was based on the necessity to update a Web application for handling traffic incident information that used unsuitable user interfaces, which generated unnecessary navigational elements,

The initial system used script technologies in obsolete server systems, as generated from AWK and UNIX Shell commands.

The traffic incident information is intended for road users who wish to plan their routes prior to leaving, or during their trip to verify the road status and to check possible alternatives routes in case of new incidents.

In addition, to be useful for individual road users, this information would be used by transportation companies who would plan the most economic and fastest routes for their drivers

The interaction with data was initially by means of a text file, made available on the Internet. However, the need for a more efficient access mechanism was identified.

Another requirement was a proposal for a way to handle traffic incident information, because there was not a common vocabulary available for the different applications

It was necessary to find a way to integrate the different access mechanism (WAP- and I-Mode telephones PDA's and voice access through conventional telephones) with a single integrated system and not by means of separate web applications.

At the same time, another important requirement was the necessity to generate an interface in at least the following languages: English, German, French, Italian, Portuguese and Spanish.

9.3. INITIAL STATE OF THE SYSTEM

The initial WAP system for handling the handling of traffic incidents was characterized by the following limitations:

- A user interface with a number of unnecessary screens, with inefficient application navigation mechanisms and no update of static information due to the lack of maintenance.

- Deficiency of graphical user interface elements that allows for a more "user-friendly" navigation.

- Interaction with incident information, through a public text file available at Web direction: http://www.dgt.es/gsm.txt

- Use of an obsolete technology to interact with the incident information based on AWK and UNIX shell commands.

- Complex application maintenance, due to the lack of separation between the presentation and the application logic, meaning WML embedded in the functional part of the dynamic interaction.

Figure 68 displays a graphical description of the initial system:

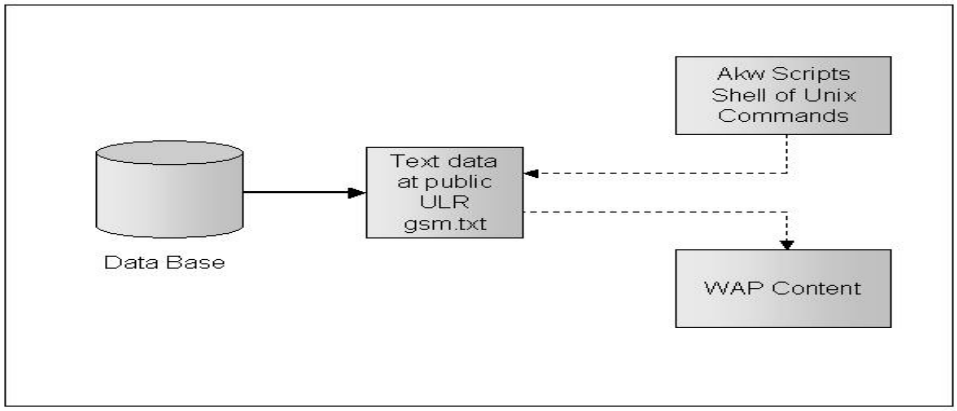


Figure 68. InitialDGT's WAP Service.

9.4. INITIAL SYSTEM ARCHITECTURE PROPOSED IN THE FIRST STAGE

The proposed system is platform independent due to the use of Java-technology. It uses Java Servlets at the server-side and interacts with available XML formatted traffic incident data. The updates of the data are the same as the text-data.

The summary of used technologies is as follows:

- Server: Linux or Solaris or Windows 2000, with Apache and Tomcat
- Database: Informix, interaction via XML
- Emulator for testing: Openwave's UPSDK versions 5.1, 6.1 and 6.2, Ericsson,
- Devices: Telephone WAP Ericsson T69i

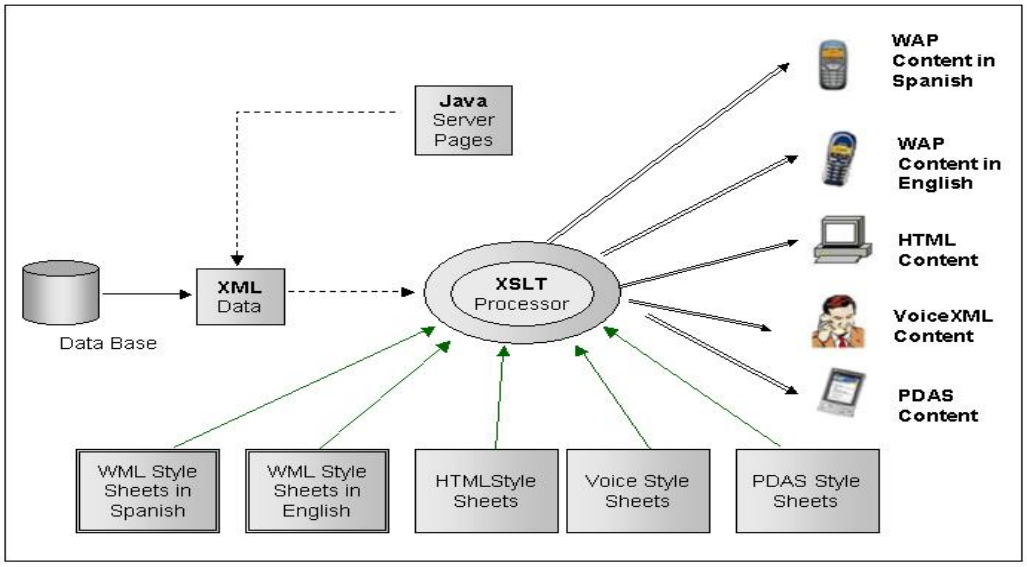


Figure 69. Proposed Architecture DGT's WAP Service.

Additionally the application navigation mechanisms were improved by eliminating unnecessary steps. Also different interface versions were created, classified mainly in:

- A version for conventional WAP devices.
- A version for devices that allow the use of graphical user interface elements.

- A version developed under the markup limitations existing in Pocket PC PDAs.
- A test version for access through voice commands.

The initially generated interface was based on the Spanish and English languages.

One of the main differences with other approaches is the improvement in the separation of content and display elements. For example the description of the menu options is stored in a pure data form in the different languages in a XML file.

The presentation is defined by means of style sheets. So, in case of a need to modify some elements, it is much easier to locate and make the changes. This update process does not affect other components.

9.5. MAIN PROBLEMS

One of the main disadvantages was the difficulty to get the selected new technology integrated with the existing computers. This was necessary because there was no funding available to replace existing hard- and software.

For this reason it was necessary to configure not only Tomcat, but also to migrate the operating system in the test machine. It was also necessary to update to a Java version required for the development (Java Development Kit),

Due to the new use of XML in the application it was necessary to find the most appropriate and efficient way to interact with a XML processor.

Different functional prototypes were developed that allowed the clarification of an efficient mechanism for multilingual and multi-device interaction necessary for the first prototypes. The large number of static documents and dynamic interaction made the update already complex.

In relationship with the tests for the developed system, the difficulty appeared in the large number of possible options that the user has and the presence of multiple languages.

One of the most complex parts of the site was related to the almost infinitive amount of possible devices that can be connected to the Internet and can have access to the application. It is economically not feasible to purchase all kind of devices, which a user can own. Also the specific limitations and differences between the emulated and real device.

In first stage of these work device independence principles was not used due to the incompatibility between the available Java/XML functionality and the server technology used. The available hardware was not able to support newer versions of the Java virtual machine. This resulted in using an older version of Java, which restricted the use of common XML-libraries.

9.6. DEVELOPMENT BASED ON THE PROPOSED FRAMEWORK

The developed web application is based on the in this thesis explained principles and proposal and has been tested with WAP, I-Mode mobile phones, iPAQ PDAs, as well as with a conventional telephones using voice control. In relation to the specific aspects related to ubiquitous web applications an analysis was made of the capabilities of all the target devices. The manufacturer data was analysed as well as different websites that summarize the capabilities.

From this basic set of information a grouping process was started. Furthermore it was determined that the diverse target devices could visualize the necessary information. The requirements for the development are based on the following formats:

- WML 1.0, which is supported by all present WAP devices (but with restricted functionality).
- WML 1.3 to give more additional support to WAP devices, supporting drop-down lists and other graphical user elements.

- CHTML to give support to the I-Mode-HTML based phoned, with or without colour capacities, to give support for different models iPAQ PDAs and VoiceXML supporting conventional telephones through voice portals.

For the handling of incident information a vocabulary was proposed to manage this information. In order to model the structure of this data it was decided to use XML schemes, considering it a mature way of describing the data and with a number of advantages over other possible alternatives, such as DTDs.

For the logical modelling was used UML . (See Figure 70)

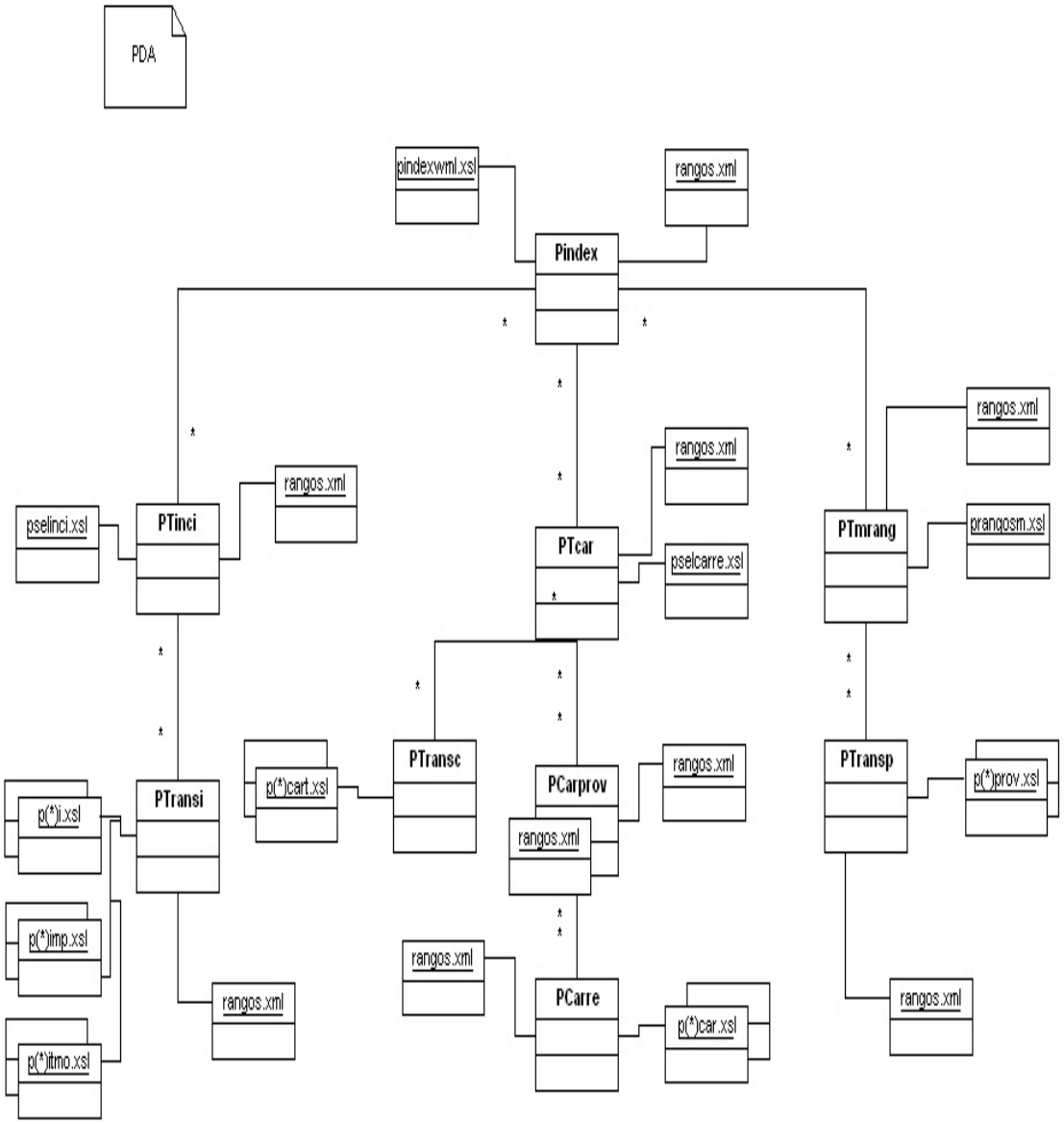


Figure 70. UML Classes for the DGT's Ubiquitous Application

The summary of the navigation between the different pages corresponding to the version for PDAs can be seen in Figure 71.

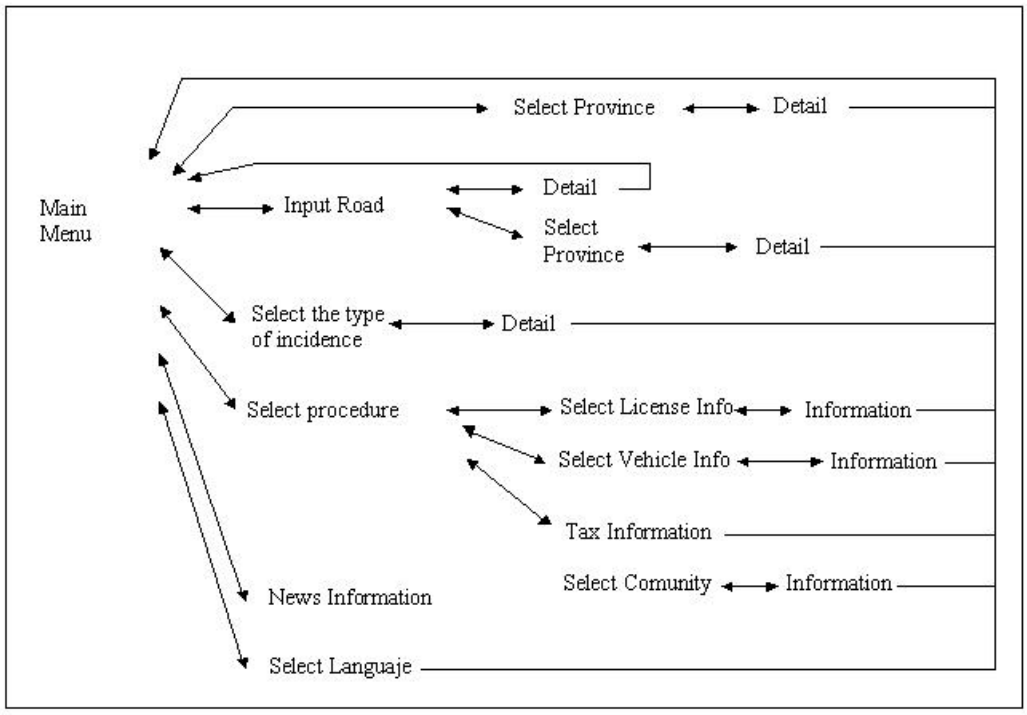


Figure 71. Diagram of Navigation for the application based on PDAs

Some of the WAP-devices do not support drop-down lists. This functionality is needed to enable users to select information based on a specific province. This problem made it necessary to modify the navigation structure including a range selection.

The diagram that summarizes navigation in this type of devices can be seen in the Figure 72.

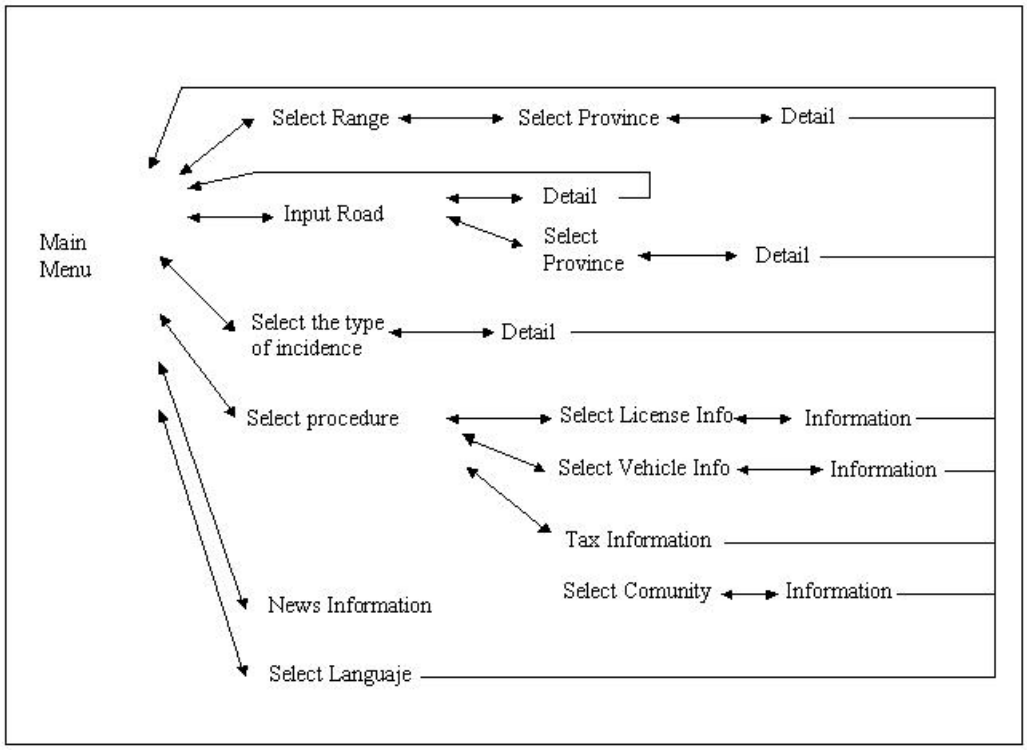


Figure 72. Diagram of Navigation for the application based on PDAs

The information is managed in a unified way by means of the previously described architecture and the contents can be accessed in different languages: English, French, German, Italian, Portuguese and Spanish.

In this project, UML was used for conceptual and logical modelling. An instance of the proposed physical modelling architecture proposed in this thesis was used. [See Figure 73]

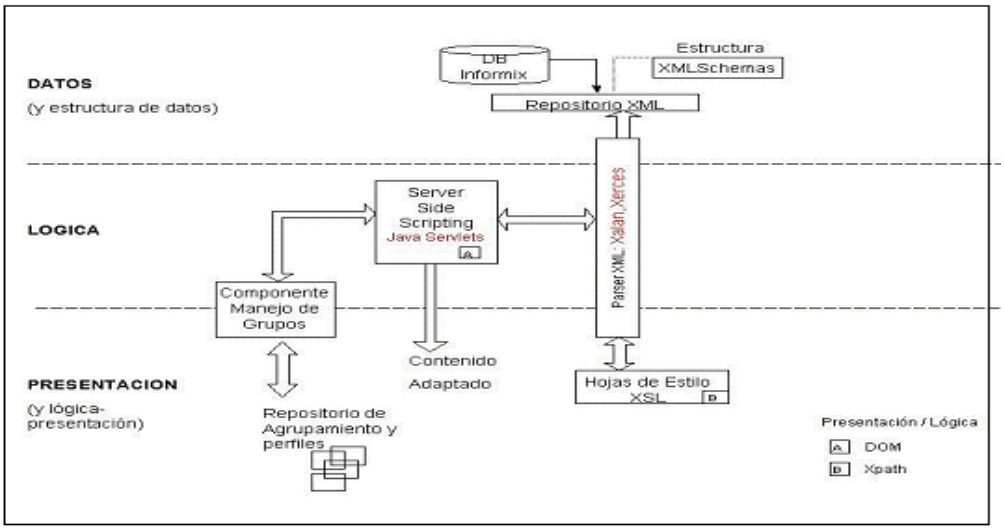


Figure 73. Instance of the proposed architecture

The server platform is based on Java Servlets running in a Tomcat web container, under a Linux operating system. Furthermore the open-source XML parser XALAN is used.

Additionally, diverse tests of migration were performed, using different environments, such as Solaris and Windows, with the purpose of verifying the viability of the application running on other platforms. It was verified that - except for some minimal configuration modifications, the developed system could be ported without problems.

Fort the authoring process of the application, one of the initial versions of the integrated environment of development proposed in this work was used. With this it was possible to verify its operation and to analyse improvements that would have to be included.

In Figure 74, a two-page Pocket PC presentation is paginated into three-page GUIs enabled WAP mobile phone, and into three pages standard WAP phone presentation.
It is observed in the PDA interface that graphic capabilities such as background images and colour text is included.

These interfaces correspond to the selection of provinces with the purpose of visualizing the existing incidences in a selected province.

Figure 74. Pocket PC PDA Interface of the application

In case the selection is made from a mobile device, the possible ranges are seen or displayed for the selection of the province name, taking into account that it's not possible to visualize all the names in one page. For example, A-BA includes the provinces which name begins with a letter within the range of A and BA.

After selecting the range, the names corresponding to provinces within that range are displayed. Once the province has been selected, a description of incidents existing in that province is displayed. (See Figure 75)

Figure 75. WAP 1.2 Interface of the application

The interface corresponding to an old WAP phone without graphic interfaces capabilities can be seen in Figure 76.

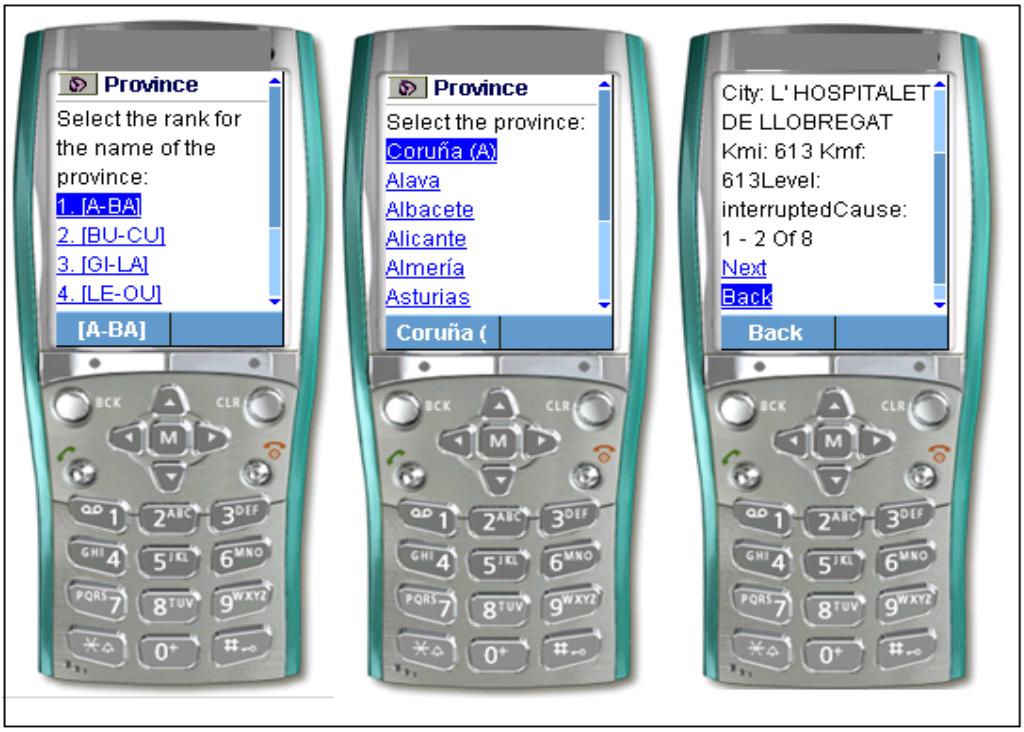


Figure 76. WAP 1.2 Interface in old WAP phones

9.7. CONCLUSIONS

From the tests made it was verified that the solution established in this thesis was used as basis for the development of a ubiquitous web application of traffic incident information. This application is based on dynamic and multilingual interaction and its performance has been tested both with emulators, mobile phones, PDAs and conventional telephones based on voice applications.

The above-mentioned proposals are characterised by the independence of server platforms and commercial technologies, which facilitates its use in different projects. The final version of the application that has been described in this chapter was implemented and has been used since February 2004.

It is possible to have real-time access to the information of traffic incidences classified by regions as well as by highways. The information is presented in five languages: English, French, German, Italian, Portuguese and Spanish, and the interface has been generated keeping in mind the style guides provided by the Spanish mobile telephone operators. With this application it has been possible to prove that the approach and guidelines included in the general framework of this thesis have been of use in real cases, which constitute a type I+D+I work (Investigation + Development + Implementation).

CONCLUSIONS

Starting from the study and analysis of the state of the art, and the research work carried out in this doctoral thesis, the following contributions have been created:

- From the analysis of the state of the art it has been determined that there are some approaches for the modelling of web applications. However, only a small amount includes context elements in their proposal. Additionally, and up to where the study got to, no proposals were found that would include the physical modelling of ubiquitous web applications, therefore there is a risk that many projects might depend only on the experience of a development team and might not have a framework as guide.

- A **framework** has been proposed to be used as guide for the development of web applications that can be accessed from many devices. This framework has been validated through its application in the development of different projects carried out by the 'Applied Transport Telematics' team at the Institute of Robotics from the University of Valencia. This framework is an alternative to the modelling proposals and to the development of ubiquitous web applications, putting the emphasis on the handling of the physical modelling. This aspect was found to be insufficiently developed in the alternatives of existing Web applications. From the tests carried out and the ubiquitous web applications created from the framework, its efficiency has been demonstrated for the interoperability and independence handling of the platform.

- A **device grouping mechanism** has been proposed. Its use avoids the development of as many applications as destination devices might be in the process of developing an application. With this mechanism device groups and their properties can be found, making it easier to develop formats (based on style sheets) for these device groups rather than individually. The proposed grouping mechanism can be used in different types of ubiquitous web applications since its development is not directly related to the inherent aspects of traffic information systems.

- An **architecture** has been proposed for the development of ubiquitous web applications that interact with different types of data sources and involve semantic elements. This architecture does not depend on a development technology on the server's side and includes a proposal a content generation process adapted to device groups. The architecture considers one device-grouping model that might be based on the mechanism proposed in this thesis or in any other way that might be developed in the future.

- An **integrated environment for ubiquitous web applications authoring** has been created. It can be used either in the authoring process of style sheets that involve including the device profile treatment and device independence principles. This integrated environment also allows for the possibility of generating scripts processing templates that will transform the style sheets to generate adapted content based on different mark-up languages.

- Also a proposal for a **vocabulary of traffic incidences** has been made. This vocabulary has been used not only in a multi-device environment but also in a multi-language environment.

- From the tests made it was verified that the solution established in this thesis was used as basis for the development of a ubiquitous web applications of Traffic Incident Information. This application is based on dynamic and multi-lingual interaction and its performance has been tested both with emulators, mobile phones, PDAs and conventional telephones based on voice applications.

- The information included in the developed ubiquitous Web portal is presented in five languages: English, French, German, Italian, Portuguese and Spanish, and the interface has been generated keeping in mind the style guides provided by the Spanish mobile telephone operators. With this application it is possible to prove that the approaches and guidelines included in the general framework of this thesis have been of use in real cases, which constitute a type I+D+I work (Investigation + Development + Implementation).

The above-mentioned proposals are characterised by the independence of server platforms and commercial technologies, which facilitates its use in different projects.

- It has been proved how the CC/PP protocol is constituted in a clear RDF application meaning a useful approach for the processing of device profiles. In relation to the possibility of using XML schemas for the device description, which only provide the profile description with syntactic elements; they are not ready for a semantic treatment as it occurs with descriptions based on RDF.

In relation to the use of ubiquitous web applications in Intelligent Transport Systems , the following has been concluded:

- Multi-device web applications and in general the access to the content of Internet is useful for intelligent transport systems since the Internet is the only worldwide and wireless media that enables people to get real-time and personalised information

- Mobile phones are already being used by millions of people in the world, far more than car navigation system vendors can hope to achieve in years. A wireless Internet mobile phone, could therefore be seen as an alternative to traditional car navigation systems. A micro-navigation device on the mobile terminal can receive a limited portion of data corresponding to the route computed and its immediate surrounding.

  However, it is not safe for a driver to be reading guidance information on a tiny screen. It would be even more dangerous than having a conversation on a mobile phone and holding the steering wheel with the other hand. Therefore, it would be better to use wireless access to the Internet with mobile devices before starting a trip. An Internet accessibe phone also allows pedestrians to ask for traffic conditions or a transport company to plan the trips of its vehicles.

In relation to the use of voice applications in Intelligent Transport Systems, the following has been concluded:

- The access to any type of information while driving will be much safer with the use of voice applications or for those vehicles that have hands-free access. A service of voice alerts and a voice application on traffic incidents allow information to be accessed about the road conditions in real time, while at the same time it is not a danger for the road users.

  In the case of the voice application, it is possible to include a wide range of information apart from the traffic incidents, such as important telephone numbers, trip advice or the nearest gas stations and restaurants. With this information, the user can plan his/her journey itinerary while driving and can avoid any potentially dangerous situation, making a good distribution of the traffic density possible.

- In this thesis, a process has been described to carry out a voice interaction with an application that generates dynamic content and that is based on VoiceXML. With this structure, it is possible to access the server contents from a telephone, both fixed and mobile, and from a computer. The development of this type of system is mainly oriented to telephone access because a computer user prefers a graphic interface to a voice interface.

- The voice applications inherit the advantages and disadvantages of technologies like the voice synthesis and voice recognition. The quality of the recognition, the background noises and the difficulty for the user to pay constant attention must be specially considered. Also in the process of text-to-speech conversion it is important to simulate human-to-human conversation as best as possible.

- Finally, we considered important the standardized adoption of vocabularies and ontologies that facilitate the interoperability of the new applications that

are developed, and their total integration with other applications available in Internet.

- From the tests made the objective sought was reached. It implied obtaining a high degree of scalability in relation to managing interfaces and the possibility of giving support to the format that is supported by new devices.

FUTURE WORK

- To carry out tests to the proposed framework in several application areas that are different to the traffic information systems

- To widen the proposed grouping mechanism allowing the use of as many properties as needed by the devices, as long as to consider the possibility of including relations among these properties.

- To extend the semantic model associated with the proposed architecture and include the existence of several semantic processing models (that are currently being developed by the International Scientific Community), and components for the intelligent processing of this type of information.

- To expand and prove these proposals with mechanisms for location based services.

- To expand the proposed integrated environment of authorting and development so that it will include the handling of components based on ontology, as well as their processing mechanisms.

- To include mechanisms that will allow concepts to be handled based on user and individual preferences of all the developed proposals. These mechanisms can be based on the automatic acquisition of the user's profile. It would therefore be interesting to evaluate techniques for machine learning, natural language processing, neural networks, cooperative filtrate and intelligent agents.

- To widen the proposed architecture in such a way that it will allow the inclusion of security elements at different levels and the maintenance of the independence philosophy of technologies on the server's side.

- To carry on with the work done in order to determine which elements must be included so that the multimodal access (several access media) to ubiquitous

applications may be taken into account. With this multimodal access, entries for instance through voice systems could be given and answers could be generated through a certain graphic interface in the device. [W3C03c]

- To evaluate the possibility of including the treatment of Semantic Mobile Web Services in the model generated in this thesis, in such a way that the ubiquitous interfaces can be integrated with register and recalling systems characteristic of the Web Systems.

- Mobile phones have been closed systems; that is, the software that shipped with the device was all that was available to the user. There was no capability to add new applications to the device or even to update existing applications. In the future, when the user of a mobile device can decide, for example, the operating system and navigator to be employed, the user must count on application development mechanisms that can be adapted to this type of changes.

GLOSSARY

3GPP:	3rd Generation Partnership Project
ADM:	Araneus Data Mode
ADO:	ActiveX Data Objects
API:	Application Programming Interface
ARTS:	Advanced Road Telematics in the South-west
ASP:	Active Server Pages
ASR:	Automatic Speech Recognition
CASE:	Computer-Aided Software Engineering
CDMA:	Code Division Multiple Access
CC/PP:	Composite Capabilities/Preferences Profile
CC/Ppex:	CC/PP Exchange Protocol
CHTML:	Compact HTML (phone)
CONNEG	Content Negotiation Working Group in the IETF
CORBA:	Common Object Request Broker Architecture
CSS:	Cascading Style Sheets
DAML+OIL:	DARPA Agent Markup Language +Ontology Inference Layer
DCOM:	The Distributed Component Object Model
DELI:	Delivery context Library for CC/PP and UAProf
DOM:	Document Object Model
DTD:	Document Type Definition
DTMF:	Dual Tone Multi-Frequency
ETSI:	European Telecommunications Standards Institute
FIPA:	Foundation for Intelligent Physical Agents
GML:	Generalized Markup Language
GPRS:	Global Packed Radio Switched
GPS:	Global Positioning System
GSM:	Global System for Mobile Communications
GUI:	General User Interface
HDM:	Hypermedia Design Method
HDML:	Handheld Device Markup Language (phone)
HTML:	Hypertext Markup Language

HTTP:	Hypertext Transfer Protocol
HTTPex:	HTTP Extension Framework
HTTPS:	Hypertext Transfer Protocol Secure
IANA:	Internet Assigned Numbers Authority
IETF:	Internet Engineering Task Force
IHTML:	I-Mode HTML
IOTP:	Internet Open Trading Protocol
JDK:	Java Development Kit
JENA:	Java Enabled RDF
LBS:	Location Based Services
LDAP:	Lightweight Directory Access Protocol
MPEG:	Moving Picture Experts Group
MIME:	Multipurpose Internet Mail Extensions
MML:	Mobile Markup Language
MMS:	Multimedia Message Service
OOH:	Object Oriented Hypermedia
OOHDM:	Object – Oriented Hypermedia Design Method
OMG:	Object Management Group
OPES:	Pluggable Edge Services
OTA:	Over The Air
OWL:	Web Ontology Language
PDA:	Personal Digital Assistant
PML:	Phone Markup Language
RDF:	Resource Description Framework
RDFS:	RDF Schema
RESCAP:	Resource Capability Protocols
RFC:	Request For Comments
RMM:	Relationship Management Methodology
RSS:	RDF Site Summary
SAX:	Simple API for XML
SDK:	Software Development Kit
SDP:	Session Description Protocol
SERTI:	South European Road Telematics Implementation
SGML:	Standard Generalized Markup Language

SHOE:	Simple HTML Ontology Extensions
SMIL:	Synchronized Multimedia Integration Language
SMS:	Short Message Service
SRGF:	Speech Recognition Grammar Format
SSL:	Secure Socket Layers
SVG:	Scalable Vector Graphics
TC:	Terminal Capability
TCN:	Transparent Content Negotiation
TCP/IP:	Transmission Control Protocol/Internet Protocol
TDMA:	Time Division Multiple Access
TTS:	Text-to-Speech
XFORMS:	XML Forms
XHTML:	Extensible Hypertext Markup Language
XML:	Extensible Markup Language
XOL:	XML-based ontology-exchange Language
XSL:	Extensible Stylesheet Language
XSL-FO:	Extensible Stylesheet Language - Formatting Objects
XSLT:	Extensible Stylesheet Language Transformation
UAProf:	User Agent Profile
UDP:	User Datagram Protocol
UML:	Unified Modelling Language
UMTS:	Universal Mobile Telecommunications Systems
URI :	Universal/Uniform Resource Identifier
URL:	Universal/Uniform Resource Locator
WDP:	Wireless Datagram Protocol
WML:	Wireless Markup Language
W3C:	World Wide Web Consortium
WAP:	Wireless Application Protocol
WHTTP:	Wireless Profiled Hypertext Transfer Protocol
WBXML:	WAP Binary XML
WML:	Wireless Markup Language
WSP:	Wireless Session Protocol
WTLS:	Wireless Transport Layer Security
WTP:	Wireless Transaction Protocol

WSP:	Wirelesses Session Protocol
XHTML:	Extensible Hypertext Markup Language
XSLT:	Extensible Stylesheet Transformations

REFERENCES

[Aki02]Akif, M., Brodhead S., Cioroianu A. Java y XML. Anaya Multimedia/Wrox. April, 2002.

[All02]Allamaraju S. Beust, Cedric. Davies J. Programación Java Server con J2EE Edición 1.3. Anaya Multimedia/Wrox. 2002.

[Ami02]Amit S., et al.. Managing Semantic Content for the Web. IEEE Internet Computing. Vol. 6, #. 4. July/August 2002. Pags. 80-87.

[And00]Anderson, C. GPRS - A Key step for the mobile Internet. Wireless Developer Network, April 2000. Available on the Internet at: http://www.wirelessdevnet.com/channels/wireless/features/gprs.html [Visited: April 19, 2003]

[Apa01]Apache Software Foundation. Apache Tomcat. Available on the Internet at: http://jakarta.apache.org/tomcat/

[Apa02]Apache Software Foundation. Cocoon. Available on the Internet at: http://xml.apache.org/cocoon/

[Atz02]Atzeni P., Mecca G., Merialdo O.. Managing Web-Based Data: Database Models and Transformations. IEEE Internet Computing. Vol. 6, #. 4. July/August 2002. Pags. 33-37.

[Aus00]Australian National University. Filling the WAP Gap. Wireless Communication Project. Available on the Internet at: http://www.tomw.net.au/2000/scspss.htmlSystem Prototype.htm [Visited April 18, 2003]

[Ber00]Berners-Lee, T. Primer: Getting into RDF & Semantic Web using N3,2000. Available on the Internet at: http://www.w3.org/2000/10/swap/Primer.html [Visited April 18, 2003]

[Ber01]. Berners-Lee T., Hendler J., and Lassila O.. The Semantic Web: A New Form of the Web Content That is Meaningful to Computers Will Unleash a Revolution of New Posssibilities. Scientific Am, Vol 284, #. 5. May ,2001.

[Ber99]. Berners-Lee T., Fischetti M., and Dertouzos M. Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its inventor. Harper, San Francisco, 1999.

[Ber01b]Bertino E., Catania B. Integrating XML and Databases. IEEE Internet Computing. Vol. 5, #. 4. July/August 2001. Pags 84-88.

[Ble01] Blesa, J.. Proyecto Fin de Carrera de Ingeniería Informática Universitat de Valencia: Editor de Páginas WAP Dinámicas. Valencia, Spain, 2001

[Bor92] Borenstein, N , Bellcore. N. F., Innosoft. MIME specification. June, 1992. Available on the Internet at: http://www.ietf.org/rfc/rfc1341.txt?number=1341.
[Visited: April 19, 2003]

[Boy03] Boyera S. Device Independence Challenges. WWW2003 - W3C Track. Budapest, May, 2003.

[But01]Butler, M. Current Technologies for Device Independence, 2001. Available on the Internet at: http://www.hpl.hp.com/techreports/2001/HPL-2001-83.pdf [Visited: April 19, 2003]

[But02]Butler, Mark, et al.. Device Independence and the Web. IEEE Internet Computing. Vol. 6, #. 5. September/October 2002. Pags 81-86.

[Cau02]Cauldwell, P., Chawla, R., Chopra, V.. Servicios Web XML. Anaya Multimedia/Wrox. 2002.

[Cer02]Ceri, S., Fraternali, P., Maera M. Conceptual Modeling of Data-Intensive Web Applications. IEEE Internet Computing. Vol. 6, #. 4. July/August 2002. Pags. 20-30

[Cer03]Cervera, A.. Proyecto Fin de Carrera de Ingeniaría Informática: Servicios Web basados en Voz para el Servei Catalá de Tránsit. Valencia, February, 2003.

[Che00]Chen G., Kotz D., "A Survey of Context-Aware Mobile Computing Research," Technical Report, TR2000-381, Dept. of Computer Science, Dartmouth College, November, 2000.

[Com01]. Comisión de Transportes, Colegio de Ingenieros de Caminos, Canales y Puertos. Libro Verde del Transporte en España. April, 2001.

[Con00] Conallen J.. Building Web Applications with UML. Addison Wesley, 2000.

[Coo00]Cook J. XML Sets Stage for Efficient Knowledge Management.IEEE IT Professional. May/June 2000. Vol. 2, #. 3. May 2001. Pags 55-57.

[Cun01]Cunliffe, D.. Developing usable Web sites - a review and model. Journal: Internet Research: Electronic Networking Applications and Policy. Volumen: 10, #4. Pags: 295-308.

[Dat02]Datta, A., Kaushik, D., Thomas, Helen,et al. Accelerating Dynamic Web Content Generation. IEEE Internet Computing. Vol. 6, # 5. September/October 2002. Pags. 27-36.

[Des01]Destounis, P, et. Al.. Measuring the mean Web page size and its compression to limit latency and improve download time . Journal: Internet esearch: Electronic Networking Applications and Policy. Volumen 11, # 1. Pags: 10-17 2001.

[Dey99] Dey, A., Abowd, G. Towards a Better Understanding of Context and Context-Awareness, GVU Technical Report GIT-GVU-00-18, Graphics, Visualization and Usability Center, Georgia Institute of Technology, 1999

[Doc01]DoCoMo. I-Mode. Available on the Internet at: http://www.nttdocomo.com/ [Visited April 20, 2003]

[Doc02]Docomo. I-Mode. Available on the Internet at: http://www.nttdocomo.co.jp/english/i/tag/I-Modetag.html [Visited: April 20, 2003]

[Doc00]DoComo. DoCoMo'S I-Mode.Toward Mobile and Multimedia in 3G. Proceedings of the Forty-Seven Internet Engineering Task Force Adelaide, Australia. March 29, 2000. Available on the Internet at: http://www.ietf.org/proceedings/00mar/slides/plenary-imode-00mar/sld001.htm [Visited: April 19, 2003]

[Dor01]Dornan A.. The Essential Guide to Wireless Communications Applications, From Cellular Systems to WAP and MCommerce. Prentice Hall, USA, 2001.

[Edg01]Edgar B. The VoixeXML Handbook. Understanding and Building The Phone Enabled Web. CMP Books. 2001.

[Eri04]. WAP architecture overview. Ericsson. Available on the Internet at: http://www.ericsson.com/mobilityworld/sub/open/technologies/wap/about/wap_architecture_overview. [Visited: April 18, 2004]

[Eur01] European standardization in Intelligent Transport Systems. . DATEX traffic and travel data dictionary (version 3.1a). ENV 13106. 2001

[Fen01] Fensel D and Musn, M..The Semantic Web: A Brain for Humankind. IEEE Intelligent Systems. 2001.

[FIP02]. FIPA Device Ontology Especification. 2002. Available on the Internet at: http://www.fipa.org/specs/fipa00091/XC00091C.pdf.

[Fox02] Fox G.. XML and the Importance of Being an Object. IEEE Computing. Vol. 4, # 3. May/June, 2002. Pags 96-98.

[Gam03]Gamboa, J. M., López, J.M,, López, L.F.. Proyecto Fin de Carrera de Ingeniería de Sistemas Universidad Autónoma de Bucaramanga: Estudio aplicado del desarrollo de portales Web con acceso a través de voz. Bucaramanga, Colombia, February, 2003.

[Gar00]García L., Rodero L. Proyecto de Fin de Carrera Ingeniería Informática Universidad de Valladolid: Desarrollo de Aplicaciones Vocales en VoiceXML, Universidad de Valladolid, Spain, 2002.

[Gar03] Garzotto F., Schabe D., Paolini P. HDM-A Model Base Approach to Hypermedia Application Design. ACM .Transaction on Information Systems, Vol. 11, #1, 1993.

[Gil01]Gil H.. Data Over Wireless Network, Bluetooth, WAP & Wireless LANs. McGrawHill. USA, 2001.

[Gom01] Gómez J., Cachero C. y Pastor O.. Conceptual Modeling of Device-Independent Web Applications, IEEE Multimedia, 2001.

[Gue02]Guerrero S., Pabón Rodríguez G.. Proyecto Fin de Carrera de Ingeniería de Sistemas Universidad Autónoma de Bucaramanga: Portal Web Para Acceso a través de dispositivos móviles. Bucaramanga, Colombia, 2002 .

[Gru93]. Gruber T.R. A Translation Approach to Portable Ontology Specification. Knowledge Acquisitions, Vol.5, 1993.

[Hil01]Hillborg M.. Wireless XML Developer´s Guide. McGraw-Hill Osborne, February, 2002.

[Hon04] Honkela, Timo . Description of Kohonen's Self-Organizing Map. Available on The Internet at: http://www.mlab.uiah.fi/~timo/som/thesis-som.html [Visited April, 2004]

[HP01]Hewlett Packard. HP DELI. Available on the Internet at: http://delicon.sourceforge.net [Visited: April 19, 2003]

[IBM01]IBM. Web Intermediaries (WBI). Available on the Internet at: http://www.almaden.ibm.com/cs/wbi/ [Visited: April 19, 2003]

[IBM02] IBM WebSphere Transcoding Publisher 4.0 Specification. Available on the Internet at: http://www-306.ibm.com/software/pervasive/transcoding_publisher/ [Visited: March 19, 2004]

[Isa01] Isakowitz, T. ; Stohr E. y Balasubramanian P.. R.MM: A Methodology for Structured Hypermedia Design. Communications of the ACM, 1995.

[Jai01]Jaideep R., Anupama R.. XML: Data's Universal Language. IEEE IT Professional. Vol. 2, # 3. May/June, 2000. Pags 32-36.

[Jon01]Jones, B. G., "Web Browser Tends and Technologies", Language Learning and Technologies, Vol 4., #. 1, May, 2000, pp.6-11. Available on the Internet at: http://llt.msu.edu/vol4num1/emerging/default.html [Visited April 20, 2003]

[Kap97]Kaplan, R.. M. Intelligent Multimedia Systems: A Handbook for Creating Applications Wiley, John & Sons, Incorporated. 1997.

[Ker02]Kerer, C., Kirda,E., Kurmanowytsch, r.. A Generic Content-Management Tool for Web Databases. IEEE Internet Computing. Vol. 6, # 4 July/August, 2002. Pags. 38-42.

[Kle01]Klein M.. XML, RDF, and Relatives. IEEE Intelligent Systems. Vol. 16, # 2. March/April, 2001. Pags. 26-28.

[Kot01]Kothari, B., Claypool, M.. Dynamic Web pages: performance impact on Web servers Journal: Internet Research: Electronic Networking Applications and Policy. Volumen 11, # 1, Pags 18-25. 2001.

[Mae01] Maedche, Alexander. Ontology Learning For the Semantic Web. Kluwer Academic Publishers. February, 2002

[Mar01] Maruyama H., et. al. XML and Java, Developing Web Applications. Second Edition .Addison-Wesley, 2002.

[Mec99] Mecca G., et al. The Araneus Guide to Web-Site Development,tech. report. Universidad de Roma. Roma, March. 1999.

[Med01]MediaLabs. Mobile Search Engines. http://www.medialab.sonera.fi/workspace/MobileSearchEnginesWhitePape.pdf [Visited: April 19, 2003]

[Mel01]Melnik, S.k. "RDF API Draft". The RDF API homepage. January, 2001. Available on the Internet at: http://www-db.stanford.edu/~melnik/rdf/api.html [Visited: April 19, 2003]

[Mic01] Microsoft. Mobile Internet Toolkit OverView. http://www.gotdotnet.com y http://www.asp.net. [Visitado March 8, 2004]

[Mic02]Microsoft. Microsoft. Tested Devices. http://msdn.microsoft.com/vstudio/device/mobilecontrols/devices.aspx. [Visited: December 17, 2003]

[Mil98] Miller E., "An Introduction to RDF", Online Computer Library Centre, Inc., Dublin, Ohio, May, 1998. Available on the Internet at: http://www.dlib.org/dlib/may98/miller/05miller.html [Visited: April 19, 2003]

[Mur01]Muratore, F. UMTS, Mobile Communications for the future. Editorial John Wiley &Sons. February, USA, 2001.

[Nam02]Nambiar U., et al. Current Approaches to XML Management. IEEE Internet Computing. Vol. 6, # 4. Julio/August 2002. Pags 43-51.

[Nij95]. NijKamp, P. Pepping, G. Banister, D. Telematics and Transport Behaviour. Springer. 1995.

[Noy01]. Noy N., et. al. Creating Semantic Web Contents with Protegé-2000. IEEE Intelligent Systems. 2001.

[OMA02]Open Mobile Alliance's UAProf Specification. 2002. Available on the Internet at: www.wapforum.org/what/technical.htm [Visited: April 19, 2003]

[Pan03]. Pan J. and Horrocks I. RDFS(FA) and RDF(MT) Two Semantics for RDFS. ISWC-2003 .2003

[Pas99] Pascoe J, Ryan N. Mobile Computing in a Fieldwordk Environment. Available on the Internet at: http://www.cs.kent.ac.uk/projects/mobicomp/Fieldwork/index.html August, 1999 [Visited: April 19, 2003]

[Pix01]PIXO. Micronavegador Pixo para I-Mode. Available on the Internet at: http://www.pixo.com/ [Visited: April 19, 2003]

[Pou01]. Poupinel, J.F. Mossé, O. Intelligent Mobility. Intelligent Transport Systems Europe (ERTICO). Available on the Internet at: http://www.ertico.com/INDEX.htm. 2001. [Visited: April 19, 2003]

[Rab01]Rabinovitch,E. J."Wireless Internet". Available on the Internet at: http://www.comsoc.org/ci/public/2000/jun/ciint.html [Visited: April 19, 2003]

[Rat00]Ratliff, John.DoCoMo As National Champion: I-Mode, W-CDMA and NTT'S Role as Japan's Pilot organization in Global Telecommunications. Department of Sociology, Santa Clara University. September 13, 2000. Available on the Internet at: http://tprc.org/abstracts00/docomopap.pdf [Visited: April 18, 2003]

[Ros97]Rossi G.;Schwabe D. y Garrido A. Design Rule in Hypermedia Application Development. Proceeding 8th ACM Conf Hypertext. ACM Press, 1997.

[Sat99] Schmidt A., ET. AL. Advanced interaction in context. In Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC'99, pages 89-101, Karlsruhe, Germany, September 1999. Springer Verlag.

[Sch01]Schwinger, W. Doctoral Thesis: Modelling Ubiquitous Web Applications.Univesity Johannes Kepler, Linz, Austria. 2001.

[Sch94] Schilit B., Adams, N. & Want, R. Context-aware computing applications. Proceedings of the 1st International Workshop on Mobile Computing Systems and Applications, Los Alamitos, CA: IEEE, 1994.

[Sel02]Seligman L., Rosenthal A.. XML's Impact on Databases and Data Sharing. IEEE Computing. Vol. 34, # 6. Nov, 2002. Pags 59-67.

[Sha02]Sharna , Chetan, Kunnis, Jeff. VoiceXML Strategies and Techniques for Effective Voice Application Dvelopment with VoiceXML 2.0. Willey Computer Publishing, 2002.

[Son02] Sonera Media Lab. Mobile Search Engines. November 15, 2002.. vailable on the Internet at: http://www.medialab.sonera.fi/workspace/MobileSearchEnginesWhitePape.pdf [Visited: April 19, 2004]

[Sow99]Sowa, J., Dietz D.. Knowledge Representation Logical Philosopiocal and Computational Foundations. Cole Publishing Co., Pacific Grove, CA, 1999.

[Sta02]Staab S. Ontologies' Kisses in Standarization. IEE Intelligent Systems, 2002.

[Sun01]Sun Microsystems. JSP: Java Server Pages. Available on the Internet at: http://java.sun.com/products/jsp/ [Visited: April 19, 2003]

[Thu01]Thuraisingham, B., Thuraisingha, B.. XML Databases and the Semantic Web. CRC Press. March,2002.

[Vol04]Volantis Systems. Available on the Internet at: http://www.volantis.com [Visited: March 8, 2004]

[Wad01]Waddington, P. Gill J. Oracle9i Application, Server. Wireless Edition in Action. Oracle Magazine, Jan. - Feb. 2002.

[Wal02] Wallace P., et. al. I-Mode Developer's Guide. Addison-Wesley, 2002.

[Wuw01]Wuwongse, et. Al. XML Declarative Description: A Language for the Semantic Web. IEEE Intelligent Systems, 2001.

[X901]X-9. Imimic Emulator for I-Mode. Available on the Internet at: http://www.x-9.com/mimic/beta.asp [Visited: April 19, 2003]

[Zaw01]Zawinski, J.. HTTP request header field / http/1.1 Accept Header Fields. March 1998. Available on the Internet at: http://www.mozilla.org/build/user-agent-strings.html [Visited: April 19, 2003]

W3C RELATED STANDARDS

[W3C02a]W3C. Authoring Scenarios for Device Independence. Note. July, 2002. Available on the Internet at: www.w3.org/2001/di/public/as/ [Visited: April 19, 2003]

[W3C98]W3C. CHTML. Available on the Internet at: http://www.w3.org/TR/1998/NOTE-compactHTML-19980209/ [Visited: April 19, 2003]

[W3C99a] W3C. User side Framework for content negotiation. W3C Note, Julio 1999. Available on the Internet at: http://www.w3.org/TR/NOTE-CCPP/

[Visited: April 19, 2003]

[W3C03a]W3C. Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies W3C. Working Draft, 25 March, 2003. [Visited: April 19, 2004]

[W3C99b]W3C. Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation.Note. July 1999. Available on the Internet at: http://www.w3.org/TR/NOTE-CCPP/ [Visited: April 19, 2003]

[W3C99c]W3C. CC/PP exchange protocol based on HTTP Extension Framework. (Note). June, 1999. Available on the Internet at: http://www.w3.org/TR/NOTE-CCPPexchange . [Visited: April 19, 2003]

[W3C00a]W3C. Composite Capabilities/Preference Profiles: Terminology and Abbreviations. Working draft. July 21, 2000. Available on the Internet at: http://www.w3.org/TR/CCPP-ta/ [Visited: April 19, 2003]

[W3C00b]W3C. Composite Capabilities/Preference Profiles: Requirements and Architecture. Working draft. July, 2000. Available on the Internet at: http://www.w3.org/TR/CCPP-ra/ [Visited: April 19, 2003]

[W3C00c]W3C. Composite Capabilities/Preference Profiles (CC/PP): Structure. Working draft. July, 2000. Available on the Internet at: http://www.w3.org/TR/2000/WD-CCPP-struct-20000721/ [Visited: April 19, 2003]

[W3C00d]W3C. CC/PP Attribute Vocabularies. Working draft. July, 2000. Available on the Internet at: http://www.w3.org/TR/2000/WD-CCPP-vocab-20000721/ [Visited: April 19, 2003]

[W3C02b]W3C. CSS Mobile Profile 1.0. W3C Candidate Recommendation, 25 de July, 2002. Available on the Internet at: http://www.w3.org/TR/css-mobile [Visited: April 19, 2003]

[W3C00e]W3C. CSS media types. Available at: www.w3.org/TR/REC-CSS2/media.html [Visited: April 19, 2003]

[W3C02c]W3C. Delivery Context Workshop. March, 2002. Available on the Internet at: www.w3.org/2002/02/DIWS [Visited: April 19, 2003]

[W3C02d]W3C. Delivery Context Overview for Device Independence. W3C Working Draft 13 December 2002. Available on the Internet at: http://www.w3.org/TR/2002/WD-di-dco-20021213/. [Visited: April 19, 2004]

[W3C02e]W3C. Device Independent Authoring Techniques workshop. Available on the Internet at: www.w3.org/2002/07/DIAT/ [Visited: April 19, 2004]

[W3C04a]W3C.Authoring Techniques for Device Independence. W3C Working Group Note, February, 2004 [Visited: April 8, 2004]

[W3C03b]. W3C. Device Independence Principles. W3C Working Group Note 01. September 2003 [Visited: April 19, 2004]

[W3C00f]W3C. HTTP 1.1. 2001. Available on the Internet at: HTTP/1.1 <http://www.w3.org/Protocols/>.

[W3C00g]W3C. HTTP-ex. Available on the Internet at: http://www.w3.org/Protocols/HTTP/ietf-http-ext/ [Visited: April 19, 2003]

[W3C99d]W3C. List of Checkpoints for Web Content Accessibility Guidelines 1.0 Available on the Internet at: http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/checkpoint-list.W3C Recommendation, 1999. [Visited: April 19, 2003]

[W3C02f]W3C Mobile Group. DICE (Device Independent Content Engine). Available on the Internet at: http://dice.ccpp.info/. 2002 [Visited: April 19, 2003]

[W3C03c]W3C. Multimodal Interaction Requirements. W3C NOTE , January 8, 2003.Available on the Internet at: http://www.w3.org/TR/mmi-reqs/ [Visited: April 19, 2003]

[W3C04b]W3C.RDF Primer. W3C Recommendation 10 February 2004. Available on the Internet at: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/ [Visited: April 8, 2004]

[W3C04c]W3C. RDF Vocabulary Description Language 1.0: RDF Schema,W3C Recommendation. February, 2004 Available on the Internet at: http://www.w3.org/TR/rdf-schema/ [Visited: April 8, 2004]

[W3C02g]W3C User Agent Accessibility Guidelines 1.0. W3C Recommendation. December. 2002. Available on the Internet at: http://www.w3.org/TR/WAI-USERAGENT/ [Visited: April 19, 2003]

[W3C03d]W3C. OWL Web Ontology Language Use Cases and Requirements. W3C Recommendation. February, 2004. Available at: http://www.w3.org/TR/webont-req/#usecase-ubiquitous [Visited: April 8, 2004]

[W3C99e]W3C. XML Path Language (XPath) 1.0 W3C Recommendation. November, 1999. Available on the Internet at: http://www.w3.org/TR/xpath.
[Visited: April 19, 2003]

[W3C99f]W3C. XSLT. XSL Transformations (XSLT) Version 1.0. Recommendation. Available on the Internet at: http://www.w3.org/TR/xslt. 16 November 1999. [Visited: April 19, 2003]

[W3C00h] W3C Technical Reports and Publications. Available on the Internet at: http://www.w3.org/TR/ [Visited: April 19, 2003]

APPENDICES

APPENDICE 1. WHAT IS THE EUROPEAN DOCTORATE?

Available in Spanish version at:
http://sestud.uv.es/c/contenidos.asp?nivel=11&cont=272

On the session from the 8th November 1994, the Senate in this University agreed to pass the initiative of the Liaison Committee in the Conferences for Rectors and Presidents at this Univeristy for Members of the European Community and create the European doctorate. In order to obtain such academic qualification, the doctoral dissertations should fulfil the following requirements:

1. Positive reports of two university professors belonging to two states of the European Community other than Spain.

2. One member of the committee of the doctoral dissertation should belong to a university of a country in the European Community that is not Spain.

3. A part of the doctoral dissertation should be written in an official language of the European Community but it should not be one of the official languages in the Valencian Community.

4. The doctoral student should write a part of the thesis in a State of the European Community that is different from Spain and should stay for a period of time not lower than three months. In order to opt for such modality of academic qualification, the doctoral student should have the qualifying documents and the proposal of the Committee with the indication "European Doctorate".

Appendices

APPENDICE 2. ARTS AND SERTI PROJECTS

ARTS (Advanced **R**oad **T**elematics in the **S**outh-west) are a Euro-Regional project funded by the European Commission (DGVII-Ten-T budget). It started in November 1997. It is the first attempt from the Public Traffic Administrations in Spain (DGT and DT), France and Portugal at studying how the co-ordination of traffic telematic installation projects at regional, bilateral and multilateral levels can improve the continuity and quality of the services to be offered to the road users. This project covers the so-called ARTS corridor that links the traffic coming from Portugal with that from Europe and that mainly uses the South Atlantic axis.

The final objective is the total removal of borders for traffic purposes. Any road user should not feel the difference between the traffic managements that the various administrations use in their territory. This implies that continuity of services and inter-operability in the area of the project must be assured.

The aim of the ARTS project is to be able to cooperate in different projects during different years (several phases). During the first phase, the main objective has been to get a stable framework for cooperation where all the traffic projects in the area that could have international implications are studied and implemented harmoniously.

ARTS domains:

Traffic management & control.
Traveller information services
Freight & Fleet management
Incident & emergency handling
Monitoring infrastructure
Traffic Centres
Horizontal issues

ARTS involves three countries and different organizations:

COUNTRY	**ENTITIES**
SPAIN	- Dirección General de Tráfico (DGT) Spanish Traffic Authority (except Basque Country and Catalonia) - Dirección de Tráfico del Gobierno Vasco (DT) Traffic authority in the Basque country
FRANCE	- Direction de la Sécurité et de la circulation routières (DSCR) French Ministry of transport - Centre d'estudies techniques de l'equipement (Cete do sud-oest) Traffic Technical Laboratory South-West area - Autoroutes du sud de la France (ASAF) Toll-Motorway company -Cofiroute Toll-Motorway company - Association des societes francaises (ASFA) Toll-motorway association
PORTUGAL	Instituto Das Estradas de Portugal (IEP) Portuguese Ministryof transport

SERTI (**S**outh **E**uropean **R**oad **T**elematics **I**mplementation), as one of the six Euro-Regional projects covering multi-national regions within Europe, is part of the TEMPO programme. This framework, which has been introduced by DG-TREN (Direction General of energy and transport of the European Community), enables longer term planning and more consistent support to improve the continuity and the quality of services on the European road network.

The SERTI network covers approximately 5000km and 6 countries, and the traffic is made up of a considerable proportion of foreigners on the different national networks, with periods of heavy migrations and considerable heavy goods vehicle traffic.

Traffic management on these networks is provided by numerous organisations: More than 25 road operators and organisations dealing with traffic management in the SERTI area. What is at stake for SERTI is the improvement of the continuity and quality of information services both at national and European level.

This is why the project partners with the support of the European Commission/DG-TREN have decided to work together as part of the TEMPO programme instigated by the Commission in order to study, develop and deploy intelligent transport services in a coordinated manner on the SERTI network.

SERTI involves the following countries: Belgium, Andorra, France, Germany, Italy, Spain and Switzerland.

APPENDICE 3. PUBLICATIONS

2004

Proposal of a generalized architecture for multiplatform development of mobile and voice enabled web applications. Eduardo Carrillo, Juan José Martínez-Durá. To appear in the Sixth International Conference on Information Integration and Web-based Applications & Services (IIWAS2004), September 2004, Jakarta, Indonesia.

Proposal of an integrated computational environment for editing pervasive portals. Eduardo Carrillo, Carlos Rios, Ana Lorena Uribe, Juan José Martínez-Durá. To appear in the Sixth International Conference on Information Integration and Web-based Applications & Services (IIWAS2004), September 2004, Jakarta, Indonesia

2003

-Speakeasy. Talking Congestion. J.Samper, Arturo Cervera, Eduardo Carrillo & Ignacio Sanchez. Traffic Technology International. Editor: UK & International Press. ISBN: 1356-9252. Pags 68-70. June/July, 2003.

-Realización de un Sistema de Alertas de Tráfico por Voz en el Servei Catala de Transit. Arturo Cervera, J.Javier Samper, Eduardo Carrillo, V.Ramon Tomás. Equipamiento y Servicios Municipales, Number 106. ISBN 1131-6381. Pags: 28-31. March/April, 2003.

- A Web Tool to Use Traffic Management Plans. V.Ramón Tomás,J.F.García,J.J. Samper,E.Carrillo. Wseas Transactions on Circuits. Volumen 2..Pags 309-313. ISBN:1109-2734. January, 2003

2002

- Desarrollo de un Portal de Voz de Información de Tráfico para el Serveri Catala de Transit.J. Javier Samper, Arturo Cervera,Ignacio Sánchez, Eduardo Carrillo,Javier Martínez. III Congreso Nacional Sobre Inteligentes de Transporte: Seguridad,

Información y Tecnología. Pags: 403,410. ISBN: 84-89875-36-7. Palma de Mallorca, Spain, November 20-22, 2002.

- Dynamic Access To Web Contents Through Voice with PHP and VoiceXML. J. Javier Samper, E. Carrillo, A. Cervera, V. Ramón Tomás International Conference on Information and Communication Technologies in Education (ICTE 2002). ISBN (Vol. I) 84-95251-77-9.Pag: 527-530. Badajoz(Spain), November 13-16,2002.

- Usefulness of E-Learning at University. P. Moreno, J. Samper, Santiago Felici, Vte Ramon Tomás, E. Carrillo. International Conference on Information and Communication Technologies in Education (ICTE 2002). ISBN (Vol. II) 84-95251-78-7.Pag: 782-786 Badajoz, (Spain), November 13-16,2002.

- Proposal of Framework For Workflow Process Automation Systems Based on .NET Framework and XML Web Servies. Nascimento, R.P.C. Do; Martins, J.A.C.; Pinto, J.M.S.; Carrillo, E. System Science 2002, Applied Mathematics and Computer Science 2002, Power Engineering Systems 2002. Río de Janeiro,Brazil, October 21-24, 2002

- Integration of Voice Services in Internet Applications. Authors: Eduardo Carrillo, J.Javier Samper, Juan José Martínez-Durá. International Conference on Information Integration and Web-based Applications & Services. ISBN 3-936150-18-4. Bandug, Indonesia,September, 2002.

- Adapting Content In a Portal Of Academic Information Based On XML and PHP. Autores: E. Carrillo, S. Guerrero, G. Pabón, , J. Martínez , J.Javier Samper. Electrical and Computer Engineering Series. A series of Reference Books and Textbooks. Advances in Simulation, Systems Theory and Systems Engineering . Editor; WSEAS Press. ISBN: 960-8052-70-X, Pags 110-111. 2002.

- Development of a Solution for Editing and Compiling of Wireless Internet Applications. Autores: E. Carrillo, J.J. Samper, J.J Martinéz Durá, V. Ramón Tomás. Electrical and Computer Engineering Series. A series of Reference Books and Textbooks. Advances in Simulation, Systems Theory and Systems Engineering . Editor: WSEAS Press. ISBN: 960-8052-70-X, Páginas 107-109. 2002.

- Mobile E-Commerce Portal.Authors:E. Carrillo, S. Guerero, G. Pabón,J.Javier Samper, J. Martínez.2002 International Conference on Electronics & Hardware Systems. ISBN ISBN 960-852-65-3.Cadiz, Spain, June, 2002.
Construction Of A Tool For Dynamic Wap Applications Editing.Authors:E. Carrillo, J.J. Samper, J.J Martinéz Durá, V.Ramón Tomás2002 International Conference on Electronics & Hardware Systems. ISBN ISBN 960-852-65-3. Cadiz, Spain, June, 2002.

- Improving Traffic Management Plans Using HTML And XML.Authors:V. Ramón Tomás, J. F. García, J. J. Samper, E. Carrillo, C. CambresInternational Conference on Electronics & Hardware Systems. ISBN ISBN 960-852-65-3. Cadiz, Spain, June, 2002.

- Web Portal for access Through Mobile Devices. IV Jornadas de Investigación y Desarrollo en Telecomunicaciones, JIDTEL 2002, TECNOCOM 2002. Medellín, Colombia, April 25-26, 2002.

- Information on The Move.Authors: I. Sánchez, J. J. Samper, V. Tomás, J. Martínez, E. Carrillo,Magazine: Traffic Technology International. Pags:36-38,ISBN 13576-9252. England, January, 2002.

2001

- Migrating Web Applications To WAP.Authors:: E. Carrillo, J. Javier Samper, R., S. Felici.Third International Conference on Information Integration and Web-based Applications & Services (IIWAS 2001). Pags: 449-452.ISBN: 3-85403-1572-2, Linz, Austria, September,2001.

- Development od a WAP Information Server With Oracle 8i. Authors: J. J. Samper, E. Carrillo, R. Cirilo, P. Moreno, V. Tomás.XI Congreso Nacional de Usuarios Oracle. Lanzarote (Spain),October 24-26.

- Construction of WAP Applications With PL/SQL and Oracle 8i.Authors:: J. Samper, E. Carrillo, R. Cirilo.III Jornadas Iberoamericanas En Telecomunicaciones y Telemática. ISBN: 958-9475-19-1. Cartagena-Colombia, August 27-29, 2001.

- Comparative Analysis Of Object Oriented Distributed Models CORBA/DCOM/RMI.Authors: E. Carrillo, S. Ibarra, J. Jiménez, S. Suárez.III Jornadas Iberoamericanas En Telecomunicaciones y Telemática Organizada por : Agencia Española de Cooperación Internacional, AECI. ISBN: 958-9475-19-1. Cartagena-Colombia, August 27-29, 2001.

- Bringing Internet To The Wireless WorlD.Authors: E. Carrillo, R. Cirilo, J.J. Martínez-Durá, J.J. Samper, G.Quetglas.5th WSES/IEEE World Multiconference on Circuits, Systems, Communications & Computers (CSCC 2001). Pags: 70-74.ISBN: 960-8052-35-1. Crete-Greece, July, 2001. Selected to be published by Electrical and Computer Engineering Series. A series of Reference Books and Textbooks. Advances in Simulation, Systems Theory and Systems Engineering . Editor: WSEAS Press, 2001, ISBN: 960-8052-70-X, Pag 107-109. 2001

- Server-Side Scripting With Databases Using Asp, Php, Cold Fusion And Java Servlets.Author: E. Carrillo.5th WSES/IEEE World Multiconference on Circuits, Systems, Communications & Computers (CSCC 2001).Pags: 177-182,ISBN: 960-8052-35-1.. Crete-Greece, July, 2001. Selected to be published by Electrical and Computer Engineering Series. A series of Reference Books and Textbooks. Advances in Simulation, Systems Theory and Systems Engineering . Editor: WSEAS Press, 2001, ISBN: 960-8052-70-X, Pag 177-182. 2001

2000

- ASP: A Strategy For Web-Databases Connectivity.Author: E. Carrillo.III International School On Internet Impact on Software Engineering .INSOVINA2000.Viña del mar (Chile), June, 2000.

Integration Web/Databases In Projects Of Electronic Commerce.Author: E. Carrillo.VII Iberoamerican Workshop in Informatics,I Encuentro Iberoamericano de Cultura Digital. Cartagena-Colombia, August, 2000.

RESUMEN EN CASTELLANO

Antecedentes, Objetivos, Discusión, Conclusiones

ANTECEDENTES

La cantidad de dispositivos con capacidad de conexión a Internet crece rápidamente. En la actualidad se dispone de teléfonos móviles basados en tecnología WAP (Wireless Application Protocol) o I-Mode, Agendas Digitales Personales, Kioskos Internet, teléfonos convencionales basados en acceso a la Web por medio de la voz, dispositivos basados en televisión interactiva, electrodomésticos, entre otros.

En una reciente estadística [Vol04] se han llegado a enumerar 600 dispositivos distintos y la cifra continúa creciendo cada día. Adicionalmente se han llegado a enumerar en este estudio 200 propiedades para cada dispositivo, que incluyen entre otras lenguajes de marcado, capacidades de seguridad, capacidades de entrada, limitaciones de memoria, tipo de medio soportados, capacidades de mensajería, conectividad de red, etc.)

Antes de abordar el tratamiento de esta problemática, se describirán algunos de los términos usados en este documento.

En adelante se entenderá por aplicación web "una aplicación que puede ser ejecutada en un entorno basado en web" y por aplicación web pervasiva o ubicua "una aplicación web que puede ser ejecutada en cualquier momento, en cualquier lugar y con cualquier medio". [Sch01]

La computación pervasiva o ubicua se basa en el estudio de la convergencia entre Internet y las tecnologías avanzadas de electrónica – particularmente inalámbricas- presentes en diversos dispositivos[5].

[5] El término computación ubicua (pervasive computing) fue introducido en 1991 por el investigador de Xerox Mark Weiser.

La visión de la computación pervasiva se fundamenta en la posibilidad de conectar todo tipo de dispositivos a Internet para proporcionar información sobre cualquier cosa, en cualquier momento y en cualquier lugar.
Los dispositivos móviles que permiten el acceso a Internet incluyen un micronavegador que le permite la navegación a Internet, cuya conectividad se puede conseguir haciendo uso de diversos mecanismos (Gateways intermedios, redes inalámbricas presentes en los sistemas de telefonía móvil, mecanismos de sincronización con PCs).

El primer navegador diseñado para ser incluido en un teléfono móvil fue el Openwave UPSK, que fué inctroducido en 1977 y los primeros portales basados en accedo inalámbrico a Internet empezaron a desarrollarse en 1999, tanto para la tecnología WAP (lanzada inicialmente en Europa y Estados Unidos) como para la tecnología I-Mode (predominante inicialmente en Japón). Por su parte las PDAs fueron lanzadas al mercado a inicios de 1993, aunque sólo en el año 2000 se incluyeron capacidades de navegación en estos dispositivos. [OMA02]

En relación con los portales de voz enn Julio de 1998, General Magic lanzó un servicio que sería el padre de los actuales portales de voz. Este servicio fue revolucionario ya que únicamente haciendo uso de la voz los usuarios podían tener acceso a potentes servicios de comunicación, como e-mail, voice mail, citas, etc. L
en 1999 fue fundado el Voicexml Forum por AT&T, Lucent Technologies y Motorola, aunque previamente estas compañías habian desarrollado lenguajes de marcado para acceso por voz en forma independiente. [Sha02]

Las aplicaciones ubicuas tienen diversos elementos diferenciadores en relación con las aplicaciones tradicionales. Aspectos como la movilidad que esta asociada a las aplicaciones de este tipo y las limitaciones y diversidad de características existentes en los dispositivos, constituyen elementos de contexto que las diferencian plenamente.

Desarrollar una versión de una aplicación web para cada uno de los dispositivos con conectividad a la Web resulta inviable. Por otra parte, desarrollar aplicaciones web

que puedan ser visualizados en forma apropiada y aprovechando al máximo las capacidades del dispositivo se constituye en una tarea compleja.

Si bien, en la actualidad existen diversas propuestas para modelado lógico y conceptual de aplicaciones Web que pudieran ser expandidas al modelado de aplicaciones pervasivas, no existe una claridad suficiente en dichas propuestas para abordar el modelado físico.

A comienzos de la década del 90 predominaba el paradigma de la computación personal. Algunas consideraciones que se tenían en cuenta incluían las formas de adaptar las presentaciones de la aplicación a diferentes resoluciones de pantalla. Adicionalmente se requería un proceso de instalación de la aplicación, el cual en muchas ocasiones incluida una fase de capacitación.

Posteriormente, con el advenimiento del acceso a aplicaciones a través de la Web, se tuvieron que tomar en cuenta nuevas consideraciones, como el uso de navegadores y la organización de la información por medio de enlaces basados en el concepto de hipertexto. Adicionalmente las aplicaciones debían ser diseñadas en forma tal que pudieras ser utilizadas por cualquier usuario sin capacitación previa.

Los requerimientos del modelado de aplicaciones web han sido caracterizados por [SchO1]. Adicionalmente se han desarrollado diversas propuestas para modelado de aplicaciones web dentro de las que se destacan: HDM (Hypermedia Design Method) [Gar03], RMM (Relationship Management Methodology) [Isa01], Arenus [Mec01], OOHDM (Object – Oriented Hypermedia Design Method) [Ros01], WebML[Cer01] y OOH [Gam01].

Sin embargo, el tratamiento del paradigma de navegación por hipertexto de un documento no constituye un cambio suficiente para abordar el problema del modelado del aplicaciones que puedan ser accedidas desde dispositivos, debido a que estos contienes pantallas pequeñas y en general muy diversas capacidades, e incluso aparecen cada día diversos dispositivos, por lo que las aplicaciones web deberían estar preparadas para su ejecución en este entorno cambiante y limitado.

Tratando de abordar esta problemática, esta tesis ha surgido dentro del Grupo de Telemática Aplicada al Transporte del Instituto de Robótica de la Universitat de Valencia. En este grupo se tiene como una de las actividades el desarrollo de investigación aplicada basada en tecnologías emergentes llevada al campo de los sistemas de Transporte y Tráfico.

El desarrollo de esta tesis forma parte las actividades relacionadas con Sistemas de Información Web para los macroproyectos ARTS (Advanced Road Telematics in the South-West) y SERTI (South European Road Telematics Implementation), cofinanciados por la unión europea.

OBJETIVOS

Por lo anterior los objetivos que se plantea esta tesis son los siguientes:

Objetivo General

- Proponer un framework, entendido como un marco de trabajo genérico, que sirva como guía para el desarrollo de portales web pervasivos que puedan ser accedidos desde múltiples dispositivos, evitando el desarrollo de un portal por cada uno y teniendo en cuenta las grandes variaciones pueden existir en sus capacidades

Objetivos Específicos

- Plantear un modelo de agrupamiento de dispositivos, que permita definir una serie de grupos, así como las características asociadas a los mismos, en forma tal que puedan generarse posteriormente los formatos asociados a estos grupos de dispositivos y no a elementos individuales.

-Proponer una arquitectura de referencia para el desarrollo de aplicaciones pervasivas, que no genere dependencia de tecnologías de servidor, y que permita incorporar la solución de agrupamiento planteada previamente.

- Plantear un entorno integrado de desarrollo que facilite el proceso de edición de los documentos que sean enviados a los dispositivos y que pueda ser integrada con diferentes tecnologías de desarrollo de scripts y configuraciones de servidor, y que incluya la posibilidad de generar scripts que gestionen el proceso de transformación requerido para adaptar diferentes formatos.

- Validar la propuesta que se genere teniendo como punto de partida la interacción con información de incidencias de tráfico, en diferentes idiomas y accesible desde teléfonos móviles, PDAs y teléfonos convencionales.

DISCUSIÓN

El problema que se ha abordado en esta tesis tiene como punto de partida el hecho de que el Internet de hoy ha sido diseñado para ser utilizado principalmente desde PCs. La mayoría de los contenidos asumen que el usuario accedo a la información utilizando una pantalla de alta resolución con sofisticadas capacidades gráficas.

Cualquiera que haya tratado de acceder a la Web desde dispositivos móviles como una PDA puede notar como la información de un sitio Web preparado para PCs no puede ser visualizada en forma apropiada en este dispositivo. Si se accede a esta misma información desde un dispositivo móvil será bastante probable que se visualice un mensaje de error que indique que este tipo de contenido no puede ser visualizado en el dispositivo (excepto que la aplicación web haya sido diseñada en forma independiente de dispositivo)

Los teléfonos móviles por su parte carecen de teclado y hacer un uso intensivo de entrada de datos puede ser difícil, incluso el simple hecho de escribir una dirección puede resultar una tarea complicada para un usuario inexperto.

A partir del análisis del estado del arte se ha podido observar como no existe una guía de referencia para el desarrollo de aplicaciones web pervasivas, que vaya más allá de modelos de componentes lógicos o guías de estilo para el desarrollo de interfaces, que aunque son importantes no constituyen un marco de trabajo para abordar el problema.

En este sentido este trabajo centra sus esfuerzos en el proceso de modelado físico, que puede ser incluida dentro de otras alternativas existentes de modelado lógico y conceptual o como punto de partida para una propuesta que incluya modelado lógico y físico de aplicaciones web pervasivas de información de tráfico.

Un problema adicional que se presenta en la información existente en la web actual es la falta de vocabularios comunes que faciliten la interacción entre sistemas existentes en diferentes máquinas (y por ende entre personas y máquinas), los cuáles es muchos casos han sido desarrollados utilizando lenguajes de marcado específicos que no definen ningún tipo de semántica que vaya más allá del simple etiquetado.

Por lo tanto, las preguntas de investigación que han surgido son las siguientes:

- Qué mecanismos o propuestas pueden ser desarrollados pueden ser desarrollados con el fin de realizar un manejo eficiente de la información de incidencias de tráfico en el desarrollo de aplicaciones Web pervasivos?

- Es posible contar con un marco de trabajo de referencia que ayude a abordar el problema de la generación de contenido independiente de dispositivo y que genere propuestas que no creen dependencia de soluciones específicas de vendedor y que sean escalables?

CONCLUSIONES

A partir del estudio y análisis del arte y del trabajo de investigación realizado en esta tesis, se han generado los siguientes aportes:

- A pesar de la importancia que tienen las técnicas de modelado de aplicaciones, a partir del análisis del estado del arte se ha podido determinar que existen algunas aproximaciones para modelado de portales Web y de las existentes solo unas pocas incluyen elementos de contexto en su propuesta. Adicionalmente, y hasta donde se llegó con el estudio no se encontraron propuestas que incluyeran el modelado físico de aplicaciones web pervasivas, con lo que se corre el riesgo de que muchos

proyectos dependan solamente de la experiencia de un grupo de desarrollo y no tenga un marco de trabajo como guía.

- Se ha planteado un ***framework*** que sirve de guía para desarrollo de portales Web que puedan ser accedidos desde múltiples dispositivos. Este framework ha sido validado por medio de su aplicación en el desarrollo de diferentes proyectos desarrollados en el grupo de Telemática Aplicada al transporte del Instituto de Robótica de la Universidad de Valencia. El framework constituye una alternativa o a las propuestas de modelado y desarrollo de aplicaciones pervasivas, haciendo especial énfasis en el manejo del modelamiento físico, aspecto que se encontró como no suficientemente desarrollado en las alternativas de modelado conceptual de aplicaciones Web existentes. A partir de las pruebas realizadas y de las aplicaciones web pervasivas creadas a partir del framework se ha podido demostrar su eficiencia para el manejo de interoperabilidad e independencia de plataforma.

- Se ha propuesto un **mecanismo de agrupamiento** de dispositivos a partir de definición de prioridades en sus propiedades, cuya utilización permite evitar el desarrollo de tantos portales como dispositivos destino se tengan en el proceso de desarrollo de un portal. Por medio de este mecanismo se pueden hallar grupos de dispositivos y las propiedades de los mismos, facilitando en esta forma el desarrollo de formatos (basados en hojas de estilo) para estos grupos de dispositivos y no en forma individual. El mecanismo de agrupamiento propuesto puede ser utilizado en diferentes tipos de aplicaciones web pervasivas ya que su desarrollo no se encuentra directamente relacionado con aspectos inherentes a los sistemas de información de tráfico.

- Se ha planteado una **arquitectura** para el desarrollo de aplicaciones web pervasivas que interactúen con fuentes de datos de diversos tipos e involucren elementos semánticos. Esta arquitectura no depende de una tecnología de desarrollo del lado del servidor, e incluye una propuesta de tratamiento de la generación de contenido adaptado a grupos de dispositivos. La arquitectura considera un modelo de agrupamiento de dispositivos, que puede basarse en el mecanismo propuesto en esta tesis o en cualquier otro mecanismo que se desarrolle en un futuro.

- Se ha generado un **ambiente integrado de edición de aplicaciones web pervasivas** que puede ser utilizado en el proceso de edición de hojas de estilo que involucren la inclusión de tratamiento de perfiles de dispositivo y de principios de independencia de dispositivo. Este entorno integrado permite adicionalmente la generación de plantillas de procesamiento de scripts que realicen transformaciones a las hojas de estilo para generar contenido adaptado basado en diferentes lenguajes de marcado.

- Así mismo, por medio de la validación realizada a las propuestas realizadas, se ha realizado una propuesta de un **vocabulario de gestión de incidencias de tráfico.** Este vocabulario se ha utilizado en un entorno no solo de múltiples dispositivos sino de múltiples idiomas.

- A partir de las pruebas realizadas se pudo verificar como la solución generada en esta tesis sirvió como base para el desarrollo de una aplicación web pervasiva de Información de Incidencias de Tráfico. Este portal esta basado en interacción dinámica y multilingüe, y su funcionamiento ha sido probado tanto con emuladores como con teléfonos móviles, PDAs y teléfonos convencionales basados en portales de voz.

Las propuestas descritas anteriormente se caracterizan por la independencia de plataformas de servidor y de tecnologías comerciales, con lo que se facilita su utilización en diferentes proyectos.

-Se ha podido probar como el protocolo CC/PP se constituye en una aplicación clara de RDF y constituye una aproximación útil para el procesamiento de perfiles de dispositivo. En relación con la posibilidad de utilizar XML esquemas para la descripción de dispositivos se puede notar como si bien estos últimos son más simples de manejar, solo proporcionan elementos sintácticos a la descripción de perfiles, pero no están preparados para un tratamiento semántico, como si ocurre con las descripciones basadas en RDF.

En relación con el uso de aplicaciones web pervasivas en Sistemas Inteligentes de Gestión de Tráfico se ha podido concluir lo siguiente:

- Las aplicaciones web pervasivas y en general el acceso a contenido de Internet es útil para los Sistemas Inteligentes de Información de Tráfico debido a que Internet es el único medio de covertura mundial y con capacidad para conexiones inalámbricas en el que los usuarios pueden conseguir información personalizada en tiempo real.

Los dispositivos móviles ya están siendo usados por millones de personas en el mundo, mucho más de los usarios que podrán existir en los sistemas de navegación incluidos en vehículos en muchos años. Por lo anteriom un teléfono móvil puede ser visto como una alternativa a los sistemas de navegación incorporados en los vehículos.

Con la micronavegación disponible en los terminales móviles un usuario puede recibir una porción limitada de información correspondiente a la ruta y a los sucesos o incidencias existentes en carretera.

Sin embargo, no es seguro que un conductor lea información mientras conduce en una pantalla pequeña. Esto podría resultar aún más peligroso que mantener una conversación en el teléfono móvil y mantener el volante con la otra mando. Por lo tanto, el mejor uso que se le puede dar al acceso a internet desde dispositivos móviles es antes de iniciar un viaje.

Así mismo un dispositivo hablitado para Internet puede ser de gran utilidad a peatones que estén planeando iniciar un viaje, o a compañías de transporte que planeen los viajes de sus vehículos.

En relación con el uso de Portales de Voz en Sistemas Inteligentes de Gestión de Tráfico se ha podido concluir lo siguiente:

- El acceso a cualquier tipo de información mientras se conduce es más seguro si se utiliza una interacción basada en voz, como por ejemplo el uso de portales de voz o de dispositivos de tipo manos-libres. En esta forma, un servicio de información de incidencias de tráfico que utilice interacción por voz permite el acceso en tiempo real sin representar peligro para el conductor

Teniendo acceso a esta información actualizada, un usuario puede planear su itinerario mientras conduce y evitar situaciones potencialmente pelogrosas, y promoviendo además una buena distribución de la densidad de tráfico.

- En esta tesis, se ha descrito la forma en que una aplicación web dinámica de información de tráfico puede ser accedida por medio de interacción con voz a partir de desarrollos basados en el lenguaje de marcado voicexml. Con este enfoque es posible acceder a un servicio por medio de un teléfono convencional o móvil, aunque este tipo de sistemas está principalmente orientado a usuarios del servicio telefónico, con lo que adicionalmente se puede conseguir una mayor cobertura del servicio a este tipo de usuarios y a aquellos que tengan algún tipo de discapacidad como los ciegos.

- Los portales de voz heredad las ventajas y desventajas y desventajas de tecnologias como la síntesis y reconocimiento de la voz. La calidad del reconocimiento, el ruido de fondo y la dificultad de mantener la atención del usuario deben ser especialmente consideradas. Adicionalmente en el proceso de conversión texto-voz resulta importante simular la entonación humana en la mejor forma posible.

Finalmente, se debe considerar la adopción de vocabularios y ontologías estandarizadas (o la creación de vocabularios propios) que faciliten la interoperabilidad de nuevas aplicaciones que sean desarrolladas, así como la integración total con otras aplicaciones existentes en Internet.

A partir de las pruebas desarrolladas se puede notar como se superaron los retos planteados en los objetivos. Las propuestas planteadas en la tesis permiten un alto grado de escalabilidad, dando soporte a la aparición de nuevos dispositivos.