Hypermedia design for the mobile era

Dionysios G. Synodinos*

National Technical University of Athens,
Network Management Center,
Heroon Polytechniou 9, Zografou 157 80, Greece
E-mail: dsin@noc.ntua.gr
*Corresponding author

Paris Avgeriou

Software Engineering Competence Center (SE2C),
University of Luxembourg, 6, rue Richard Coudenhove-Kalergi,
L-1359 Luxembourg-Kirchberg, Luxembourg
Fax: +352 43 21 24 E-mail: paris.avgeriou@univ.lu

Abstract: The daunting task of designing hypermedia applications for multi-channel access by heterogeneous devices has been the target of great attention by the academia and the practitioners' community. The plethora of client platforms poses many challenges, due to the diversity of the available capabilities and restrictions of these platforms. This paper presents an attempt for designing a conceptual model for hypermedia applications that allows for easy update and alteration of its content as well as its presentation and also allows for deployment in various mobile platforms.

Keywords: hypertext/hypermedia; hypermedia architectures; hypermedia models; multi-channel access to hypermedia; mobile internet; Wireless World Wide Web.

Reference to this paper should be made as follows: Synodinos, D.G. and Avgeriou, P. (2004) 'Hypermedia design for the mobile era', *Int. J. Mobile Communications*, Vol. 2, No. 3, pp.271–284.

Biographical notes: Dionysios G. Synodinos is serving as Senior Web Engineer at the Network Management Center of the National Technical University of Athens (NTUA), Greece. He has served as the Vice Chair of the IEEE Student Branch at NTUA, Webmaster of the Board of European Students of Technology (B.E.S.T.) and President of the NTUA Local B.E.S.T. Group. He has participated in a number of European Union R. & D. projects regarding mobile hypermedia and he is the father of the "Write once 'n' Deliver Anywhere" architecture (WOnDA). His research interests concern the area of hypermedia with emphasis on architecture, multi-channel access and the semantic web. He also conducts research in patterns and object oriented programming.

Dr. Paris Avgeriou is an ERCIM (www.ercim.org) Research Fellow at the Software Engineering Competence Center (SE2C), University of Luxembourg. He received a diploma (MSc) in Electrical and Computer Engineering (1999), as well as a Ph.D. in Software Engineering (2003) from the National Technical University of Athens (NTUA), Greece. He has worked as a visiting Lecturer at the Department of Computer Science (Jan. 03 – Jan. 04), University of Cyprus, and as a research and teaching assistant at NTUA (Sep. 99 – Dec. 02). He has participated in a number of European Union R&D projects and has published several papers in international journals and conferences. He acts as a reviewer in journals, standards and conferences and is a founding member of the World-Wide Institute of Software Architects and the president of its Greek chapter. His research interests concern the area of software engineering and particularly software and business modeling, with emphasis on software architecture. He also conducts research in patterns, quality assurance, web engineering and mobile clients. His research is mainly applied in the domain of e-learning.

1 Introduction

During the last years there has been a lot of hype about the Wireless World Wide Web (W4), where hypermedia applications can be accessed by wireless clients and especially mobile users. Emerging technologies and the penetration of mobile computing in our daily lives, have led to the need for mobile access to common websites. The technology behind the mobile access to the internet usually draws upon a subset of HTML, like the cHTML (compact HTML) of NTT DoCoMo's i-mode with its millions of users in Japan, or is XML-derived like WML (Wireless Markup Language), the markup language of WAP, heavily deployed mainly in Europe.

The anticipation of faster and cheaper W4 that the 3rd Generation (3G) wireless networks and more sophisticated mobile devices will bring, results in a growing number of organisations that plan to deploy parts of their traditional websites for mobile users. This task though can be overwhelmingly difficult since several problems have arisen, concerning building and maintaining hypermedia applications for mobile users.

To begin with, there is a vast demand for the adaptation of content into a growing number of presentational templates, each one of them suitable for a different device. Only in the case of WAP-enabled phones, must the developer provide numerous templates that comply with the potential user's device capabilities and restrictions. These can be device characteristics like the screen size, e.g. a site should be deployed in different ways for a Nokia 7110 with its 96×46 pixels screen and for an Ericsson RS with its generous 360×120 pixels display. Resolution and available bandwidth are also issues. For instance, the site author should make provisions for slow GSM access, faster GPRS networks or even lightning fast 3G and beyond. Furthermore, the input peripherals bring in another degree of freedom, ranging from the standard numerical pad to Nokia's 9210 PC-like keyboard. To make matters worse, mobile clients might need different versions of web pages as a side effect of the different level of support for WMLScript, the WAP client-side scripting language. In situations like these, updating content and performing version control can become tremendously resource-consuming if not impossible.

Above all, web pages have evolved to the point of becoming too complex with all the inline client scripts and style rules in order to facilitate the ever-growing and often conflicting demands for enhanced usability and impressive 'look and feel'. Therefore the daily task of updating content can no longer be performed by a novice in markup languages, but instead designated professionals with a solid background in web authoring and a clear understanding of the architecture of the specific site must be utilised.

The question now is, given the XML technology, how can a site be engineered in order for it to achieve modifiability, maintainability and portability. Specifically, the problem that hypermedia application authors have at hand, comprises the following secondary problems:

- how can one maintain the site content by updating it, at will, without requiring him or her to master the underlying technology of presentation style sheets, client-side scripts etc. for the target mobile platforms?
- how can one modify the layout, presentation and active components of the site without affecting the content associated with them?
- how can one port the hypermedia application to alternative versions for existing mobile platforms and still make provisions for future delivery platform versions?

In this paper, we attempt to solve the above problems by proposing an XML-based multi-tier model, which is established upon the separation of the actual content, active widgets, presentation rules and the page generation process. This model is a refinement of the WOnDA (Write Once, Deliver Anywhere) model [1,2] for mobile clients.

2 Literature review

There is already work in progress by the World Wide Web Consortium (W3C), under the title of Composite Capabilities/Preference Profiles (CC/PP) [3], towards serving the same content to different clients according to their profile. In particular, the CC/PP framework aims to provide a common way for clients to express their capabilities and preferences to a server that generates content. The server then uses this information to adapt the content in a way appropriate for the client's device. This approach is particularly valuable for the case of mobile devices where the variety of clients is overwhelming. The WOnDA model can be integrated with the CC/PP model, where different presentation formats can be created in an asynchronous way to satisfy the devices' profiles collected from a central or various distributed profile registries. On the other hand, WOnDA proposes that the creation of the presentation rules is done prior to user request, and has no provision for dynamic adaptation of content transformation rules according to the capabilities of the device by reading the device's profile in real time. Such a scenario, though, will be examined in future additions of WOnDA, so that it is not imperative for the developer to know the specifics of a new device in order to make the necessary provisions for it.

Another effort by the W3C is the *XForms* [4], which is an XML vocabulary intended to be integrated to XHTML [5] or SVG [6] standards. XForms provide a platform-independent markup language for online interaction between a person (through an XForms Processor) and a remote agent. XForms are viewed as the successor of HTML forms that aim at separating the data being collected from the markup of the controls collecting the individual values thus resulting in platform independence and reusability of forms. As this effort receives the necessary support from industry and compatible clients emerge, we aim to integrate it in a future version of our model.

Curious Networks is developing Multi-channel Access XML (MAXML) [http://www.curiousnetworks.com/] which aims at satisfying the growing need to deploy applications on not just one, but many access channels. Unlike many traditional development processes based on defining presentation, MAXML focuses on Interaction Oriented Development. Developers define their applications within MAXML in terms of the interactions that users may have with these applications. Through extensive usability tests, Curious Networks have created a Human–Information Interaction Model that defines standard ways of interaction between users and information, regardless of the access channel. Although this approach is valuable in the sense that it provides a layer of transparency from the underlying delivery markup, it restricts the developers' option within its core (Human–Information Interaction Model).

Another important approach of serving the same content to different clients is Several Interfaces, Single Logic (Sisl) [7]. This is an architecture and domain-specific language for designing and implementing services with multiple user interfaces. It aims at the decoupling of interface from service logic, by employing an event-based model of services that allows service providers to support interchangeable user interfaces to a single source of service logic or data. The need to provide content in a hypermedia format to a mobile client can be satisfied using the Sisl approach, in the sense that a website can be modelled and developed using standard Sisl architecture, so that it can later facilitate interchangeable user interfaces. In fact, Sisl targets a wider variety of services and deals in greater depth with issues like how to manage the fundamental differences in the nature of interaction across the spectrum of user interfaces. As a result it does not elaborate on the actual process of creating the hypermedia documents. Also, the broader spectrum of applications that Sisl aims at, causes it to grow in complexity, making its use in smaller projects, an overkill.

Another approach worth mentioning is the User Interface Markup Language (UIML) [8] that provides a highly device-independent method to describe a User Interface (UI). UIML looks at any UI from six orthogonal dimensions: the parts comprising the UI, the presentation of these parts, the content, the behaviour, the mapping of UI controls in some domain (e.g. HTML) and the business logic connected to this UI. It is implemented as a declarative XML-compliant meta-language that allows for the implementation of many UIs without learning the language or API specific for that device. The same philosophy is adopted by WOnDA, which supports content creation without mastering the underlying technology of any of the supported platforms. UIML's latest version supports multi-modal UI that can be used simultaneously and kept synchronised. For example, a UI might offer a voice and a screen based front-end and the user can at any time switch between the two interaction modes. It supports a variety of supported domains (not only Markup Languages) and provides a canonical representation of any UI that is suitable for mapping to existing languages. Due to its view of a UI as a tree whose parts can change dynamically, it is well suited for applications that need to be multilingual.

A very recent approach by Daniel Billsus et al. for mobile access to hypermedia applications is presented in [9], where the authors advocate the need for an automated approach to the adaptation of the User Interface that uses artificial intelligence and statistical techniques. The adaptation model proposed is incremental and iterative and focuses on the user's interests as depicted by his actions and not by explicit content rating. In this way, the system learns from the user's preferences and adapts the UI in a way that provides easy access to a vast amount of information. This approach is still at a very theoretical level and remains to be tested in real-world applications.

As far as the graphics that are encountered in hypermedia applications are concerned, they are usually static bitmaps in GIF, PNG or JPEG format. The use of this kind of technology in a mobile context proves problematic due to the varying capabilities of mobile devices to render and display graphics. Traditionally the developer must adapt the image file in a number of different file formats, sizes, resolutions and depths of detail in order to accommodate the different needs of the application's target audience. During the last years, the convergence of computer graphics and other technologies such as artificial intelligence is leading to the development of *Smart Graphics* [10]**,** which recognise application requirements, user characteristics, host-machine capabilities, and target usage, and adapt themselves accordingly. The smart graphics differ from conventional computer graphics in that they can be programmed to recognise the platforms on which they are working, the tasks they must perform, and network capabilities such as bandwidth. If this adaptation is performed at the clients' side, this approach is based on the assumption that the device can handle such an intensive procedure of adapting the source graphics file. At this point WOnDA's traditional approach of before-hand server side adaptation according to target device profiles proves to be more efficient for clients with smaller processing capabilities. This is the case for most mobile internet appliances.

An interesting approach in delivering hypermedia in multiple channels is the one proposed in [11], where the authors suggest the development of a *Smart Style* layer for the WWW by employing Semantic web technologies. They propose a framework that is theoretically capable of providing intelligent stylesheets by utilising three elements:

- Common vocabularies for describing delivery contexts like CC/PP.
- Intelligent transformation methods that take into account a wide variety of delivery contexts.
- The use of explicit metadata and design knowledge. It is obvious, of course, that in practice these stylesheets would have to contain a large amount of design and domain knowledge encoded in RDF Schema [12] and DAML+OIL [13].

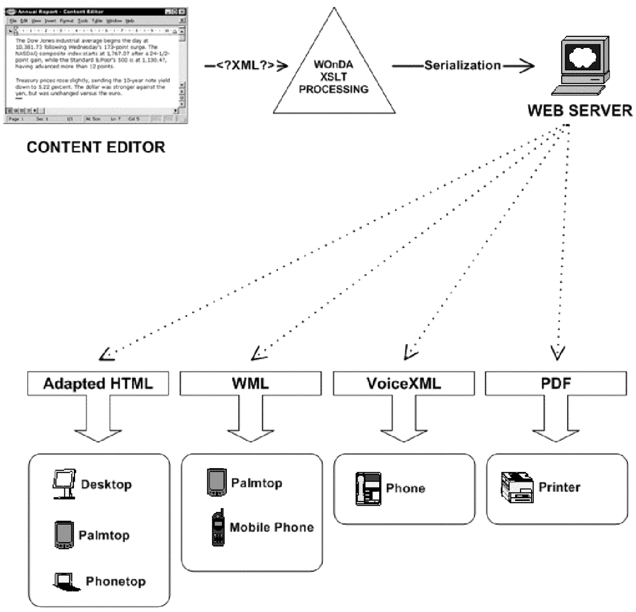
Hypermedia content is and probably always will be developed primarily for the desktop PC taking into account the capabilities and restrictions of such a device like large screens, adequate processing power, easy input mechanism etc. As mentioned earlier this practice leads to the distribution of information in a form that is usually not appropriate for the new-age internet capable devices that spring out like mushrooms these days. With this notion, the authors of [14] propose a dual-mode web browsing model that supports navigation and action in separate interfaces, serving the needs of users of smaller footprint devices. To demonstrate their model they have developed *m-Links*, a middleware proxy system that retrieves hypermedia documents and provides users with a navigation interface that separates links from page content. This 'middleman' approach for content adaptation is contradictory to our model where content is customised by the issuing server.

The issue of determining the clients' capabilities and restrictions has proved to be an overwhelming task for web application designers since the early days of the WWW. Modern trends address this issue with a variety of methods, most of them based on the interpretation of the HTTP [15] headers of the client's browser. Specifically the 'User-Agent' header is compared to a list of predefined user agents, e.g. browser software and/or client device and their characteristics. Another method is to use client-side scripting like JavaScript or VBScript. In this manner, the script can find out more about the specific settings of the environment it is called to operate on, like the resolution or colour depth of the users' display. Furthermore many server side technologies like ASP, PHP, etc. typically use a file named '*browscap.ini*' that holds information regarding the browser capabilities. Finally, commercial products have been developed to extend the functionality of this technology like the BrowserHawk (http://www.cyscape.com/products/bhawk) that aid in the development of cross-platform, browser-friendly web applications.

3 The model

Before describing the WOnDA model in detail, it is useful to examine the way in which this model is used for writing content once and delivering it to multiples clients. An overview of the models' modus operandi is illustrated in Figure 1. The content is authored in a text editor that provides XML authoring facilities in a transparent way to the author. This could be implemented as a MS Word plug-in [YAWC Pro, http://www.yawcpro.com/], an ActiveX component [XMLSpy document editor browser plug-In, http://www.xmlspy.com/download_plugin.html], or a different editor [XMLSpy IDE, http://www.xmlspy.com/products_ide.html]. This means that content can be created and updated by people with no web-authoring background, by letting them write in simple text and have it automatically converted to XML. In sequence, XSL transformations are utilised to impose style rules and presentation layout, add active objects, and generate the page in its final form, e.g. WML page, a handheld-compatible page etc. The final page is then published to a internet Server and served to the appropriate clients through the internet.

Figure 1 A macroscopic view of the model



What is the mechanism that deals with the XML files and XSL Transformations that translate raw content into a specific delivery platform? What are these XML files, how do the XSL transformations take place and what does the final result look like? These questions will be answered in the remainder of this section in the form of a guide for the construction of maintainable, modifiable and portable hypermedia applications for mobile clients.

In order to describe this mechanism we propose a conceptual model by utilising the Unified Modelling Language [16] (http://www.rational.com/uml), a widely adopted modelling language in the software industry and an Object Management Group standard [http://www.omg.org/]. Furthermore in order to define the syntax and semantics of the conceptual model we have designed a UML meta-model, i.e. a model that defines the language for expressing the conceptual model [17].

The conceptual model described here considers only static hypermedia pages and every page comprises the following elements:

- the actual content of the page that consists of text, hyperlinks, images, videos, animation etc.
- a set of navigational or promotional active objects or widgets like navigation bars, search boxes, menus, logos, ads, banners, etc.
- the general layout of the page meaning the positioning of all the above in the browser window and the rest of the markup envelope that is needed in order for the page to be syntactically valid
- hyperlinks to other hypermedia pages.

It is noted that this is a simplified and superficial model of a hypermedia page because the aim of WOnDA is not to model hypermedia applications, in general, but merely to separate content from the rest of the information and generate multiple versions of hypermedia applications. In other words the proposed model is considered to be in a lower abstraction layer compared to the usual hypermedia design models such as HDM [18], RMM [19], WebML [20], etc.

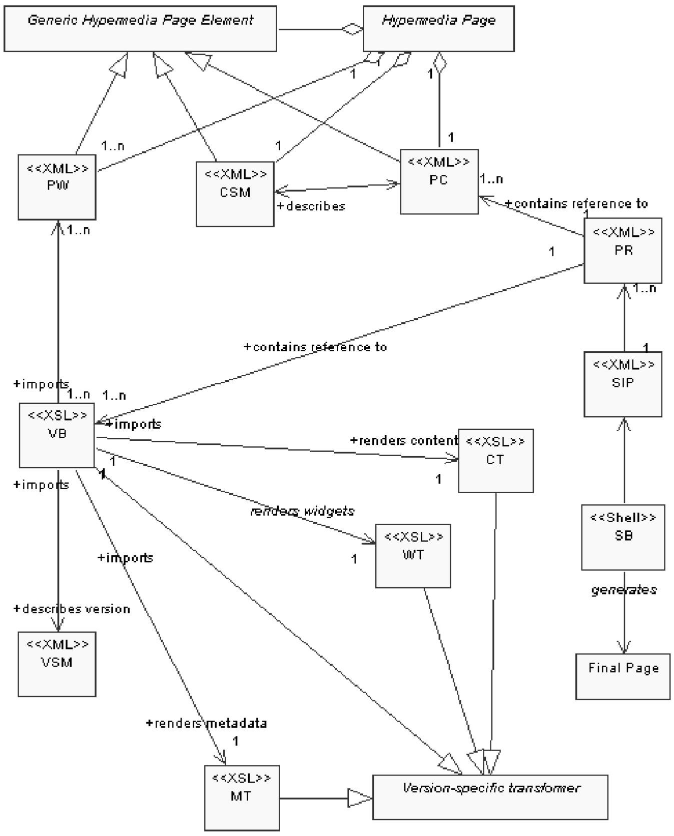
We now move on to specify the meta-model that will define the language for expressing the conceptual model. The principles of the meta-model are the following:

- The actual content (text, links and references to media files) of each hypermedia page is kept in one XML file. These files will be referred to as Page Contents (PCs). PCs represent published pages as abstract data entities without taking into account any presentation aspects derived by the desired formats.
- The task of providing content rendering information is left to a set of XSL files, which will be referred to as Content Transformers (CTs). The idea behind CTs is that if we define a set of N versions for the site under construction, every version is exactly identical to all the others in terms of textual information, since this information is provided by the PCs, but the versions differ in the layout, functionality, style and the markup that they're written in. For every one of the versions we define a CT that describes the rules necessary to transform the content provided by the respective PCs.
- In the fashion of PCs and CTs for the textual content we define Page Widgets (PWs) and Widget Transformers (WTs), which hold the necessary data and presentation rules respectively, for the widgets used.
- Metadata that are specific to the content page, as in the WML, cHTML, HTML <META> element, used throughout a version or even throughout the entire site are kept in an XML file, which will be referred to as Content-Specific Metadata (CSM).

- Metadata that are specific to a certain version, e.g. character encoding information, are kept in another XML file, called Version-Specific Metadata (VSM).
- Both content page-specific and version-specific metadata are rendered by another transformer XSL file called Metadata Transformer (MT).
- For every one of the different versions we define an XSL file, which describes the rules necessary to generate the page layout that is restricted in the context of the version. These XSL files, which will be referred to as Version Builders (VBs), do not contain any information about the rules we need to render the textual content drawn from the PCs nor the widgets used. Rather they define the general layout of the page meaning the positioning of all the above in the browser window and the rest of the markup envelope that is needed in order for the page to be syntactically valid for the corresponding presentational domain. The information about client-side scripts or additional client-side style rules (e.g. CSS), are referenced by the VB or included in it depending on the capabilities of the syntax of the relevant domain. For example, for the HTML domain this can be accomplished by the <LINK> element.

  Figure 2 depicts the relationship between the above model elements. Page Contents, Page Widgets and Content-Specific Metadata are XML files and are all specialisations of the class 'Generic Hypermedia Page Element'.

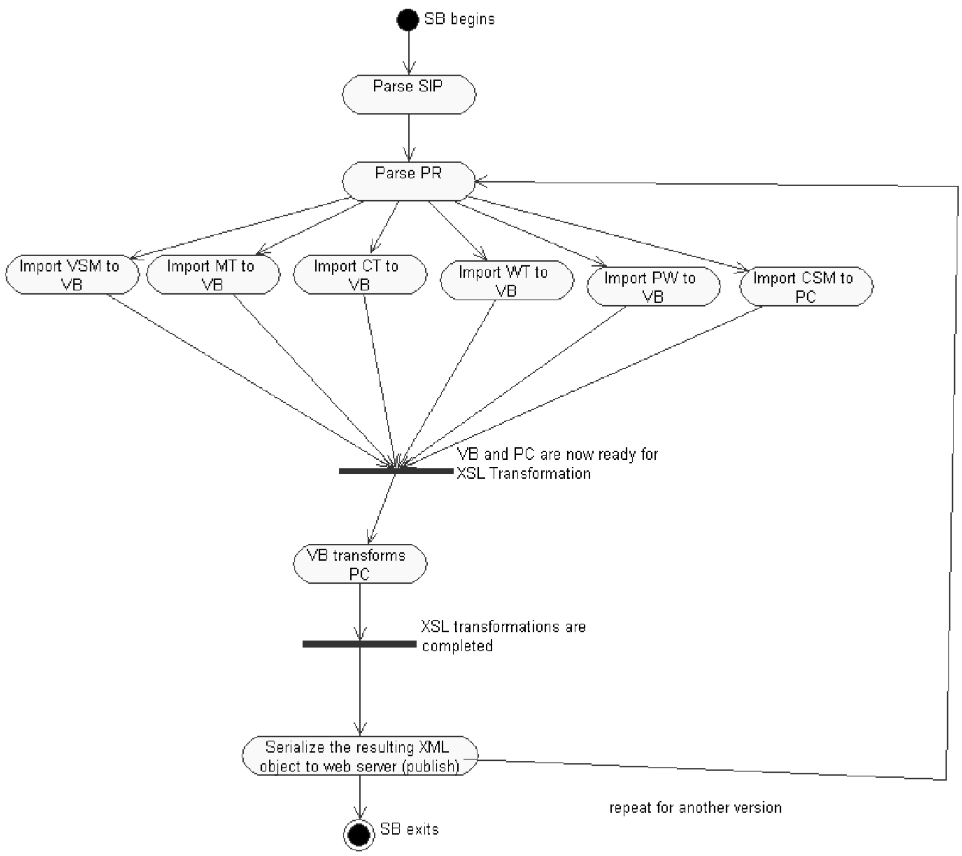
Figure 2 The hypermedia page elements and the transformers



- For every page there is a registry specific for it, called Page Registry (PR)**,** which links together all the page elements and their transformers for the various versions.
- For every site there is a main registry, the Site Index Page (SIP), which holds all the file system (or network) paths to the Page Registries.
- All the above are parsed by a processing shell, which will be referred to as *Site Builder* and provides the website administrator with a web interface to generate or update certain pages, entire versions of the site, etc.

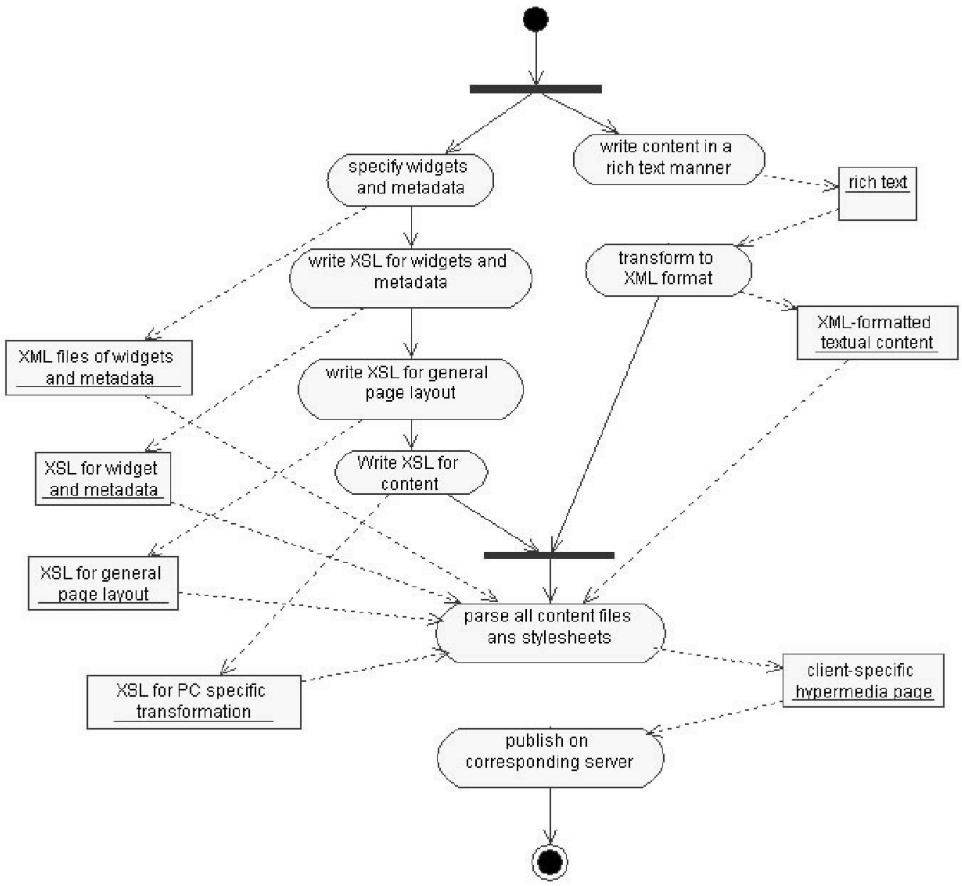
Figure 3 depicts the page creation process that takes place with the aid of the last three elements. The Page Registry gathers all the necessary data from the Page Content, the Content Transformer, the Content-Specific Metadata and the Version Builder. The Site Builder parses the Site Index Pages to look for all the Page Registries, performs the transformations and generates the hypermedia page of the appropriate format.

Figure 3 An activity diagram of the site builder functionality



The last thing to be clarified about WOnDA is the way that the model is applied. A simple process model for harnessing WOnDA is depicted in Figure 4, as a UML activity diagram, showing the discrete activities as well as the artefacts that those activities produce or take as input.

Figure 4 A process model for applying WOnDA



4 Mobile city guide of Athens: a case study

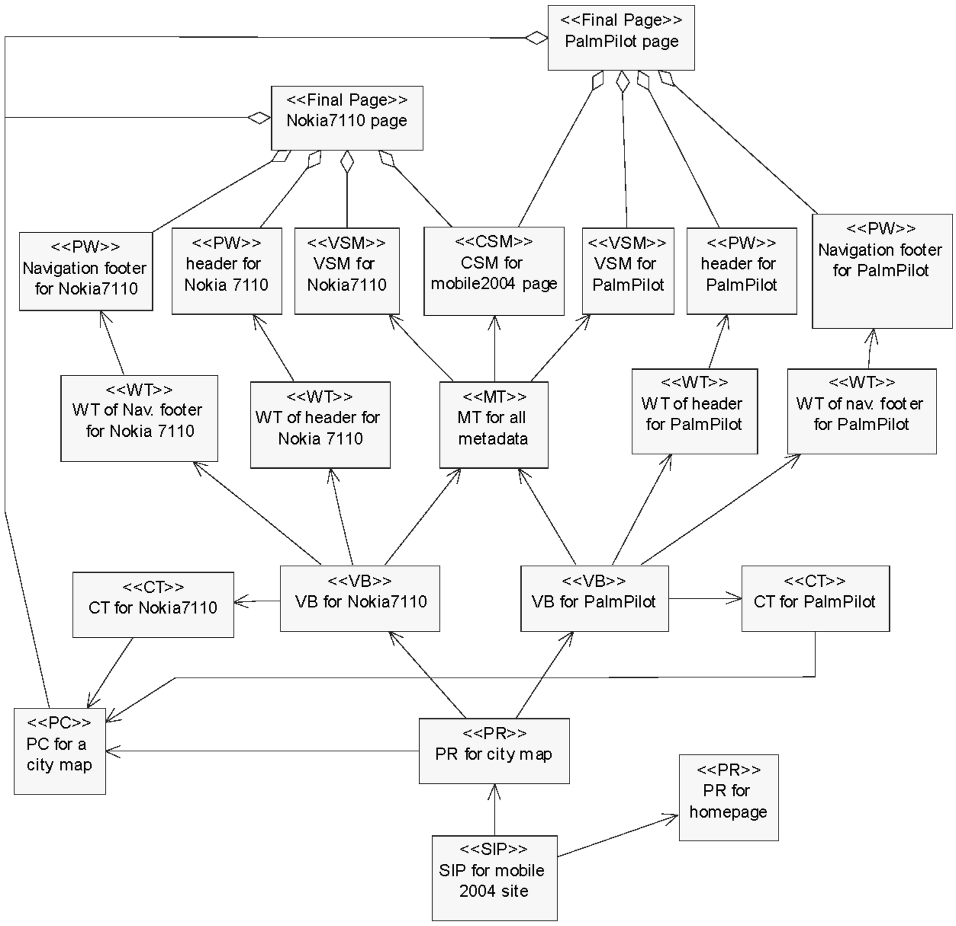
In order to examine the feasibility and effectiveness of WOnDA in a real world scenario, a demo website has been developed. The purpose of this site is to provide tourists who visit Athens for the 2004 Olympic Games and have mobile internet access, with information about the city, the Games and the various events. The aim of our effort was to provide custom access to both textual information and images e.g. maps, for a certain number of predefined mobile devices. The scenario described by the following pictures is one where four users with four different mobile clients, access the home page of the site and choose to browse a map of the city.

Figure 5 Screenshots of mobile clients

It is important to mention that since there are multiple versions of the site, one for every version available, there is a need for a mechanism that redirects any device that visits the top-level URL (e.g. www.mobile2004.gr), to the part of the site that corresponds to the device used (e.g. www.mobile2004.gr/nokia7110/index.wml). This can be easily implemented by parsing the HTTP headers of the client's original request and identifying his/her browser by the 'User-Agent' header [15]. Even in the case that the information exposed by this header does not match a client known to the system, the user can be redirected to a generic version of the site, which proves functional for his/her device, although not customised.

Figure 6 depicts the implementation of the meta-model described earlier in the case of the aforementioned demo site, focusing on the page with the city map. This page consists of a picture with the map, a header with the Athens 2004 logo and a footer with navigational widgets. This is an instance of the meta-model, i.e. a model per se that is described using all the necessary meta-model elements. The diagram shows only the elements for Nokia7110 and PalmPilot in order not to overload the diagram, even though the other two clients are represented in a similar way. Also, with the intention of keeping the diagram simple, the relationships between the elements that were defined in the meta-model are note repeated here. Every model element in this diagram is distinguished by having a stereotype defined for it. Stereotypes are a UML mechanism for extending the core of the language and are denoted by brackets in this diagram.

Figure 6 The model for the mobile 2004 site



5 Future work

Although the principles described above provide a solid foundation, they do not deal with the entirety of the diverse scenarios that are popular in contemporary hypermedia applications for mobile clients. Therefore, we plan to extend this model in various ways in order to accommodate certain features that are currently missing. First of all, dynamic content must be provided to the user in situations where the user needs to systematically draw information from a database or merely execute HTML-embedded server side scripts. Furthermore, the model needs to include provisions for multiple languages used for the same content – a strongly emerging need especially for sites like the one illustrated in the case study section.

Another issue that concerns the model per se is the use of the Object Constraint Language (OCL) [21] to define the model more formally with the use of well-specified constraints. OCL is the UML's recommended language for specifying constraints and can help in formalising the various model elements and the relationships between them.

A formal evaluation of the model's effectiveness is another step that we plan to take in order to measure the quality of this work. This is already under way with the use of the model in new hypermedia applications for mobile users.

Finally, an interesting issue that is under research is the way that WOnDA can benefit from all the work recently done in the database area in order to deal with XML documents. This is due to the variety of proposed models for storing and accessing XML information both natively and on object-relational databases.

References

1 Synodinos, D. and Avgeriou, P. (2003) *WOnDA: An Extensible Multi-platform Hypermedia Design Model*, Springer-Verlag, LNCS Vol. 2426, pp.217–228.

2 Synodinos, D. and Avgeriou, P. 'm-WOnDA: the 'Write Once 'n' Deliver Anywhere' model for mobile users', *in proceedings of Workshop on Conceptual Modelling Approaches to Mobile Information Systems Development (MobIMod'2002)*, October 2002, to be published by Springer-Verlag LNCS.

3 World Wide Web Consortium (W3C) (2000) *Composite Capabilities/Preference Profiles: Requirements and Architecture*, W3C Working Draft, 28 February, 2000.

4 World Wide Web Consortium (W3C) (2002) *XForms 1.0*, W3C Candidate Recommendation 12 November 2002.

5 World Wide Web Consortium (W3C) (2002) *XHTML 1.0 The Extensible HyperText Markup Language (Second Edition): A Reformulation of HTML 4 in XML 1.0*, W3C Recommendation 26 January 2000, revised 1 August 2002.

6 World Wide Web Consortium (W3C) (2001) *Scalable Vector Graphics (SVG) 1.0 Specification*, W3C Recommendation 04 September 2001.

7 Ballz, T., Colby, C., Danielseny, P., Jategaonkar Jagadeesany, L., Jagadeesan, R., L'aufer, K., Matagay, P. and Rehory, K. (2000) 'Sisl: several interfaces, single logic', *Journal of Speech Technology*, Kluwer Academic Publishers, 2000.

8 MarcAbrams and Constantinos Phanouriou, (1999) 'UIML: an XML language for building device-independent user interfaces', *in Proceedings of XML '99*, Dec. 1999, Philadelphia.

9 Daniel, B., Brunk, C.A., Evans, C., Gladish, B. and Pazzani, M. (2002) 'Adaptive interfaces for ubiquitous web access', *Communications of the ACM*, May 2002, Vol. 45, No. 5.

10 Smart Graphics http://www.smartgraphics.org

11 Jacco van Ossenbruggen and Lynda Hardman (2002) *Smart Style on the Semantic Web*, CWI.

12 World Wide Web Consortium (W3C) (2000) 'Resource Description Framework (RDF) schema specification 1.0.', *W3C Candidate Recommendations*, 27 March 2000, http://www.w3.org/TR

13 'Reference description of the DAML+OIL (March 2001) ontology markup language', http://www.daml.org/2001/03/reference.html

14 Schilit, B.N., Intel Research, Jonathan Trevor and Hilbert, D.M. (2002) 'FX Palo Alto Laboratory; and Tzu Khiau Koh, Xerox Singapore', *Web Interaction Using Very Small Internet Devices*, IEEE Computer, Oct 2002.

15 RFC (1945) *Hypertext Transfer Protocol – HTTP/1.0*

16 Booch, G., Rumbaugh, J. and Jacobson, I. (1999) *The UML User Guide*, Addison-Wesley.

17 The Rational Unified Process (2000) v. 2001.03.00.23, *Rational Software Corporation, Part of the Rational Solutions for Windows Suite*.

18 Garzotto, F., Schwabe, D. and Paolini, P. (1993) 'HDM – a model based approach to hypermedia application design', *ACM Transactions on Information Systems*, Vol. 11, No. 1, Jan. 1993, pp.1–26.

19 Isakowitz, T., Stohr, E. and Balasubramaniam, P. (1995) 'RMM, A methodology for structured hypermedia design', *Communications of the ACM*, August 1995, pp.34–48.

20 Ceri, S., Fraternali, P. and Bongio, A. (2000) 'Web modeling language: a modeling language for designing websites', *Proceedings of WWW9 Conference*, Amsterdam, May 2000.

21 OMG Object Constraint Language Specification, version 1.3. (1999) Framingham, MA, 1999.