The Art of Multi-channel Hypermedia Application Development

Dionysios G. Synodinos[1], and **Paris Avgeriou**[2]

(Proceedings of MobEA WS, collocated with WWW2003 conference, Budapest, Hungary, May 20th 2003)

Abstract— **The plethora of networked devices and platforms that continuously come to light, as well as the emergence of alternative ways to access the internet, have increased the demand for multi-channel access to hypermedia applications. Researchers and practitioners nowadays not only have to deal with the challenges that classic hypermedia applications pose, but also have to face numerous considerations with respect to multi-channel delivery of the applications. This paper presents an attempt for attacking the problem of multi-channel hypermedia application development. In specific it proposes a model that explicitly separates the hypermedia content from its presentation to the user through a document engineering perspective, by employing XML content storage and XSL transformations. Our work is based upon the empirical results of designing, developing and deploying hypermedia applications on multiple platforms and client devices, and on the practices of well-established hypermedia engineering techniques.**

Index Terms— **multi-channel delivery, mobile hypermedia, content personalization.**

I. Introduction

For centuries mankind has used documents for recording and distributing information. At the beginning the form in which we recorded information was necessarily the same as the form in which the documents were presented to the reader. Though the introduction of the computer introduced the potential for a more flexible handling of documents, the early digital text processing systems' purpose was only to obtain the same quality as the one produced by traditional printing techniques. The potential flexibility of the computer was mainly used to provide better support for the authoring and production process, not for the distribution and presentation process. Due to the superior quality of printed paper as compared to computer displays, the final version of a document was still disseminated to its audience in the traditional way: on paper.

With the birth of hypertext, the World Wide Web and hypermedia, researchers and practitioners were given the ability to deliver information in several alternative forms, like different versions of sites, different browsers, different client devices etc. There was a paradigm shift from 'one source - one delivery medium' to 'one source - multiple delivery media'. This new paradigm seems to expand as new technologies and new client devices emerge, like for example the wireless World Wide Web (W4), where hypermedia applications can be accessed by wireless clients, anywhere and at any time. Typical examples of alternative hypermedia access channels are NTT DoCoMo's i-mode technology [http://www.nttdocomo.com] with its millions of users in Japan and WAP [http://www.wapforum.org/what/technical.htm] that is heavily deployed in Europe and abroad.

The anticipation of faster and cheaper W4 (Wireless World Wide Web) that the 3rd generation (3G) wireless networks [http://www.3gpp.org] and more sophisticated mobile devices will bring, results in a growing number of organizations that plan to deploy parts of their traditional web sites for multi-channel access. This task though can be overwhelmingly difficult since several problems have derived in building and maintaining hypermedia applications for access by heterogeneous platforms.

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 "phonetops" and WAP-enabled phones, the developer must 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 a different way for a Nokia 7110 with its 96x46 pixels screen and for an Ericsson RS with its generous 360x120 pixels display. Also resolution and available bandwidth is an issue. 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, different clients might need different versions of web pages as a side effect of the different level of support for client-side scripting. In situations like these, updating content and performing version control can become tremendously resource-consuming if not impossible.

On top of everything else, web pages have evolved to a point of becoming too complex with all the inline client scripts and styles 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 on web authoring and a clear understanding of the architecture of the certain site must be utilized.

The World Wide Web Consortium had an early provision to such problems with the launch of XML and the related family of technologies, in order to separate the content from the rest of the information such as presentation rules, metadata, active components etc. The question now is, given the XML technology, how can a site be engineered in order for it to achieve modifiability, maintainability and portability. In specific, the problem that hypermedia application authors have at hand, is comprised of 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 and extension of the **WOnDA** (**W**rite **O**nce '**n**' **D**eliver **A**nywhere) model [1], [2] and [3] that extends the capabilities of the published model by giving the ability to maintain dynamic pages. The presentation rules are themselves separated into a set of rules for transforming the actual content, the various widgets and the layout of the pages for every one of the presentational domains. The proposed model is based on an XML repository that holds the actual (textual) content and utilizes the power of eXtensible Style Sheet (XSL) transformations in a hierarchical manner that is well suited for providing a rich set of formats for every page to be deployed in. It also facilitates easy administration, the ability to easily add new formats and fast/clear re-factoring of old ones. Also, since there is a complete separation between data and presentation, the development of content can become a streamlined process that doesn't deal with the complexity of the underlying structure of the chosen presentational domains.

II. THE MODEL

Before describing the WOnDA model in detail, it is useful to examine the way this model is used for writing content once and delivering it to multiple clients. An overview of the model's 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 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. The inclusion of references to dynamic data sources can be done in a variety of ways starting from wizard-driven predefined queries to databases for novice users, to hard-coding these references in the actual XML code for advanced users. In sequence, XSL transformations are utilized 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. Information that is not static can be dynamically generated, e.g. retrieved from a database, or a web service, and then be integrated into the page generation process. The final page is then published to a Web Server and served to the appropriate clients through the Internet.

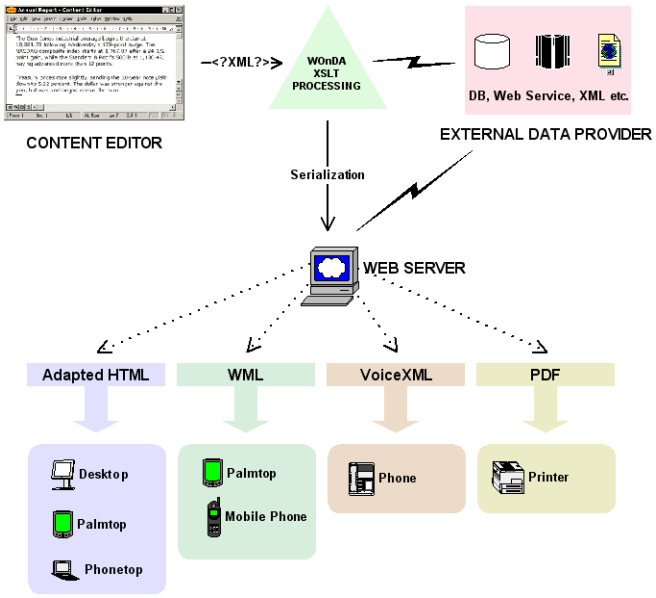

Figure 1 - A macroscopic view of the model

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 utilizing the Unified Modeling Language (http://www.rational.com/uml), a widely adopted modeling 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.

The conceptual model described here considers both static and dynamic hypermedia pages and every page is comprised

of the following elements:

1. The actual content of the page that consists of text, hyperlinks, images, videos, animation etc. as it is integrated by dynamic and static data sources (DBs, Web Services, XML files)

2. The dynamic elements of a page that will provide information at the time of the user 's request

3. A set of navigational or promotional active objects or widgets like navigation bars, search boxes, menus, logos, ads, banners etc.

4. 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.

5. 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 than usual hypermedia design models such as HDM , OODHM , RMM, WebML 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:

1. The actual content (text, links, references to media files) of each hypermedia page is described 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. They describe both static information in the form of XML segments and more dynamic information in the form of processing instructions that communicate with external modules that provide information be querying DBs, Web Services etc.

2. The task of providing content rendering information is left up 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 which describes the rules necessary to transform the content provided by the respective PCs.

3. 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. Again these entities are responsible for both static and dynamic information.

4. 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). Again this entities are responsible for both static and dynamic information.

5. Metadata that are specific to a certain version, e.g. character encoding information, are kept in another XML file, called **Version-Specific Metadata** (VSM).

6. Both content page-specific and version-specific metadata are rendered by another transformer XSL file called **Metadata Transformer** (MT).

7. 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. They rather 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 specializations of the class "Generic Hypermedia Page Element". They are also connected with an aggregation relationship with the "Hypermedia Page" class, which means that they are all part of a hypermedia page. Content Transformer and Widget Transformer are XSL files that render the corresponding Page Contents and Page Widgets. Furthermore it is obvious that Content-Specific Metadata are related to Page Contents, in the sense that metadata describe the content. Moreover, Content-Specific Metadata and Version-Specific Metadata are rendered by the Metadata Transformer. The Version Builder uses all the other transformers to render the layout of the hypermedia page and insert the appropriately transformed content, widgets and metadata into the final version-specific hypermedia page. Page Content, Page Widgets and Content-Specific Metadata, are comprised of a static part and a dynamic part. Content Transformers, Widget Transformers and the Version Builder are specializations of the "Version-Specific Transformer" class. Finally it is noted that the names of the classes "Generic Hypermedia Page Element", "Hypermedia Page", "Version-Specific Transformer" are written in italics, since they are abstract classes in this meta-model.

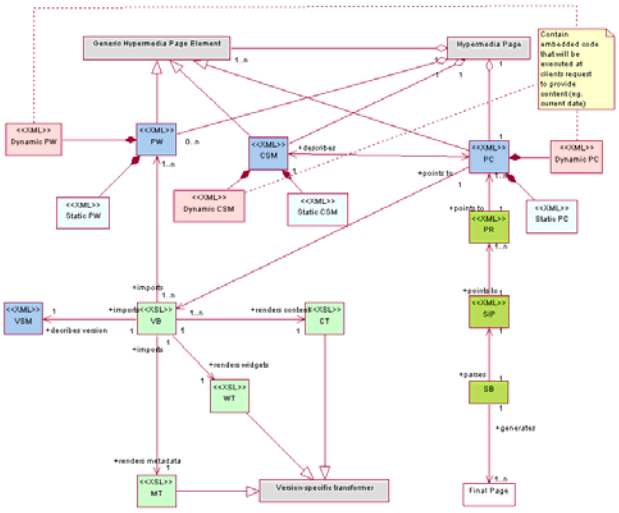

Figure 2 - The hypermedia page elements and the transformers

8. 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.

9. 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.

10. All the above are parsed by a processing shell, which will be referred to as Site Builder and provides the web site administrator with a web interface to generate or update certain pages, entire versions of the site etc.

III. 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. It is an infotainment site that provides the citizens of Athens with information regarding entertainment like movies, music, restaurants etc. 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 picture is one where a MS WindowsCE client with wireless internet access, visits the home page of the site and through a certain number of page transitions finds where the movie theater of his choice is situated on the city map. Also the final page with the map is demonstrated on a variety of other devices in order to visually demonstrate the diversity of such a mobile environment.

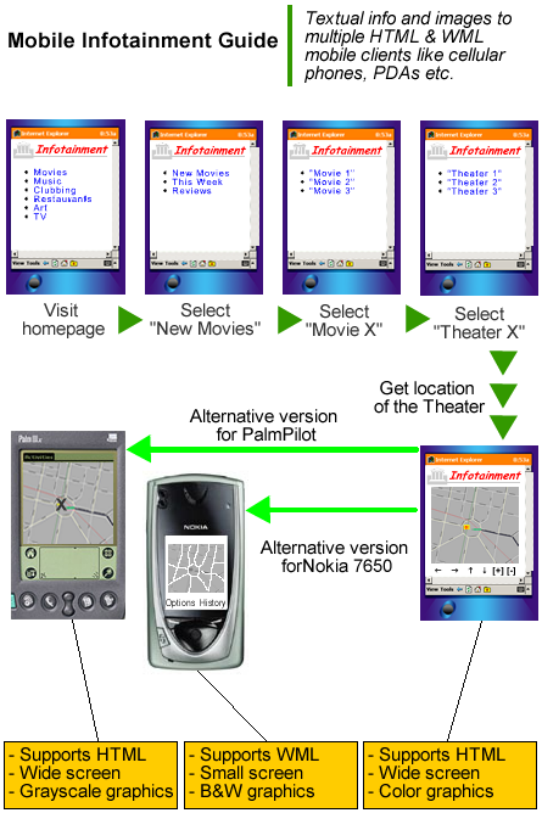

Figure 3 - Screenshots of the four different mobile clients

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.infotainmentsite.gr), to the part of the site that corresponds to the device used (e.g. www. infotainmentsite.gr/nokia7650/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. Even in the case that the information exposed by this header doesn't match a client known to the system then the user can be redirected to a generic version of the site, which proves functional for his device, although not customized. Figure 4 depicts the implementation of the meta-model described earlier in the case of the aforementioned demo site, focusing on the page with the Theater location Map in order not to overload the diagram. This page consists of a picture with the map 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. 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 guillemets in this diagram.

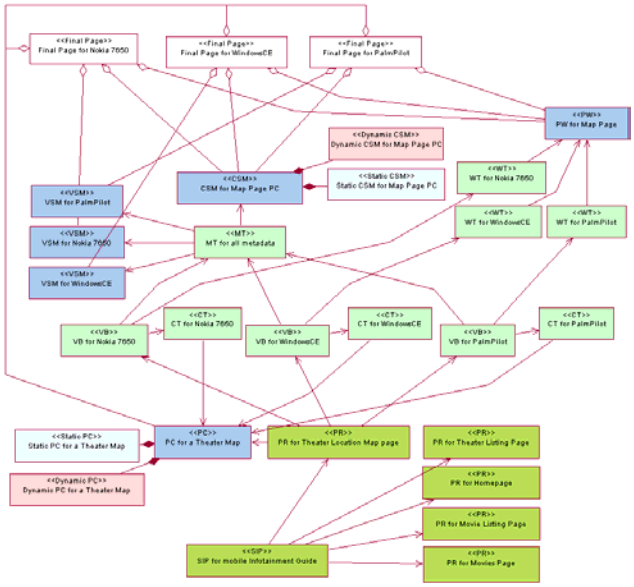

Figure 4 - The model for the mobile 2004 site

ACKNOWLEDGMENT

Dionysios G. Synodinos would like to thank the Network Management Center of the National Technical University, http://noc.ntua.gr for the valuable support he received.

REFERENCES

[1] Dionysios G. Synodinos and P. Avgeriou, "WOnDA: an Extensible Multi-platform Hypermedia Design Model", Springer-Verlag LNCS Volume 2426, pp 217-228.2.

[2] Dionysios G. Synodinos and Paris Avgeriou, "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), Tampere, Finland, October 2002, to be published by Springer-Verlag LNCS.

[3] Dionysios G. Synodinos and Paris Avgeriou, " Hypermedia Design for the Mobile Era", International Journal of Mobile Communications (IJMC), Summer 2003

AUTHORS

[1] **DIONYSIOS G SYNODINOS,**
National Technical University of Athens,
Network Management Center,
Heroon Polytechniou 9, Zografou 157 80, Greece,
Tel: +30-210-7722241, Fax: +30-210-7721866,
dsin@noc.ntua.gr

[2] **PARIS AVGERIOU**,
National Technical University of Athens,
Software Engineering Lab, Heroon Polytechniou 9, Zografou 157 80, Greece,
Tel: +30-210-7722487, Fax: +30-210-7722519,
pavger@softlab.ntua.gr