

X-SHAAD: An XML Implementation for Hypermedia Systems Modeling through SHAAD

David Mérida, Ramón Fabregat, Carlos Arteaga, and Anna Urra

Institut d'Informàtica i Aplicacions (IliA)

Universitat de Girona (UdG)

17071 Girona, SPAIN

{david.merida, ramon.fabregat, carlos.arteaga,
anna.urra}@udg.es

Abstract. In this paper we present an implementation using XML and Java (X-SHAAD) based on a modular architecture of SHAAD to support a web-based Adaptive Hypermedia System. This architecture considers the hypermedia adaptation along three different “adaptivity dimensions”: user characteristics, technology (the network and the user’s terminal), and user interaction. At present, web content delivery from server to clients is carried out without taking into account such heterogeneous and diverse characteristics as user preferences, the different capacities of the clients’ devices, the different types of access, the state of the network and the current load on the server, all of which directly affect the behavior of web services. The growing use of multimedia objects in the design of web contents further complicates the issue, greatly affecting appropriate contents delivery. Thus, the objective of the system we present here is a treatment of web pages that takes into account such heterogeneity and adapts content with the aim of improving web performance.

1 Introduction

At present, the enormous heterogeneity in terms of types and capacities of access devices, network bandwidth and the needs or preferences of users is not taken into account by a server when providing web content that is rich in images, audio or video. The server will deliver the document requested even if the terminal being used (WebTV, PDAs or mobile telephones) cannot access this content due to various limitations (e.g. display, storage capacity, processing or network access) The page is also provided independently of users’ preferences. In addition, the interaction between users in different environments (i.e. commercial, educational, etc.) modifies the requirements made on a requested web page.

To tackle this problem, different alternatives have been developed that allow universal access to any type of material while taking into account certain aspects of this heterogeneity. For example, user preferences [1][2][3]; the capacity of client devices [4][5], access types [6][7], the state of the network [8], or/and current load on the server [9][10], and/or the interaction of users[11].

We consider that adaptation can be made along three different adaptivity dimensions (Fig.1):

- *User characteristics*: preferences, background knowledge, navigation activity, etc.
- *Technology*: network state, client's device characteristics, bandwidth, etc.
- *User interaction*: collaborative, cooperative, etc.

One point in this space represents the adaptation to be made for one user, the state of their technology and their interaction with other users. In other words, it represents the adaptation as a function of three dimensions.

$$\text{adaptation} = f(\text{user}, \text{technological}, \text{interaction})$$

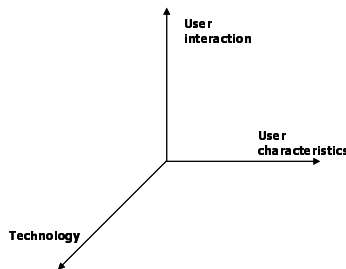


Fig. 1. Adaptativity dimensions

Cannataro in [12] and [13] proposes an architecture where the Application Domain is modeled along three different orthogonal adaptivity dimensions: *User's behavior* (browsing activity, preference, etc.), *Technology* (kind of network, bandwidth, Quality of Service, user's terminal, etc.), and *External environment* (times-spatial location, language, socio-political issues, etc.). The *External environment* dimension differs from the *User interaction* dimension of the SHAAD model. *User interaction* reflects the existing interaction between users which depends on the working environment, i.e. educational hypermedia, on-line information systems, hypermedia for information retrieval, etc. it also reflects the collaborative or cooperative aspects of this interaction.

In [14], SHAAD¹ is defined as a system that takes into account the problem of heterogeneity and tries to adapt the available information dynamically or statically and to deliver it in the most efficient way. It incorporates concepts of adaptability, adaptivity and dynamism summarized from work carried out by Brusilovsky [1] [15], and De Bra [2] [16] [17]. SHAAD represents, through its nested structure, a simple model that takes into account these concepts for hypermedia systems and reflects the necessary mechanisms for an appropriate content delivery.

¹ SHAAD is the Spanish acronym for "Adaptable, Adaptive and Dynamic Hypermedia System"

The latest modification of the system takes into account adaptivity dimensions, and is made up of 4 modules (Fig. 2):

- *Mechanisms for characterization of adaptation variables.*
- *Content repository:* data can be provided by different sources.
- *Decision Engine:* the kernel of the system and the place in which the adaptation variables and the available content are evaluated. This module reflects through three decision types the three different “adaptivity dimensions”: *user characteristics, technology, and user interaction.*
- *Content Generator:* includes the adaptation mechanisms decided upon by the Decision Engine

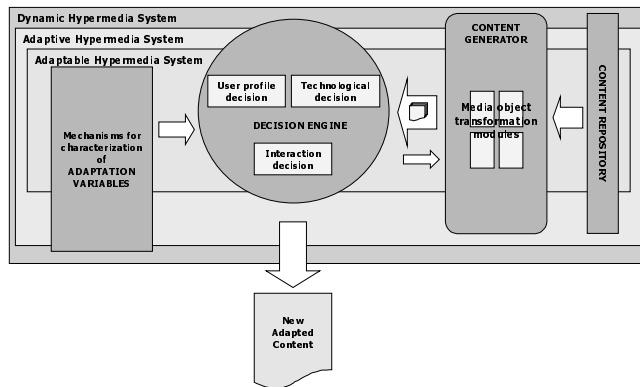


Fig. 2. Modular structure of SHAAD

The SHAAD modular structure allows an independent development of the parts it is made up of. The function of the *Content Generator* is to deliver the requested content, either through a dynamic generation (from the unit elements that make up the web page - on-line generation), or by selection from the various different static versions of this content (previously generated - off-line generation). This generator includes the media object transformation module that will make the necessary transformations of media objects.

In the present work we describe the X-SHAAD, an XML-SHAAD approach dedicated to the analysis, regeneration and adaptation of HTML content, which takes into account some of the variables of the *technological* dimension (the characteristics of the client's device and the current state of the net).

In our research, we are also developing two implementations that consider:

- the *User characteristics* dimension through MAS-PLANG (MultiAgent System PLANG²)[3]. This is a multiagent system developed to transform the virtual

² This is a Broadband Communications and Distributed System (BCDS) group project of Girona University for the “Implementation and study of a new generation telematics platform to support open distance learning.

educational environment of USD³ into an Adaptive Hypermedia System, AHS, which takes into account the student learning style; and

- the *User interaction* dimension through an Adaptive Collaborative Learning Hypermedia System [11] that relates the individual learning and the collaborative learning through an adaptive model.

This paper is organized as follows: In Section 2, we describe the implementation and processes of X-SHAAD. In Section 3, we describe how requested pages are readdressed to X-SHAAD. In Section 4 we show some adaptation examples of the system. Finally, in Section 5 we give some conclusions and describe future work.

2 X-SHAAD

X-SHAAD architecture represents one way to implement SHAAD, and the different modules defined by it are shown in Fig. 3. The processes to take into account in the X-SHAAD are:

- *HTML to XHTML transformation* at the Content Repository module.
- *Obtaining the adaptation profile to be applied* at Decision Engine module.
- *Obtaining the Adapted HTML* at Content Generator through XSLT parser.

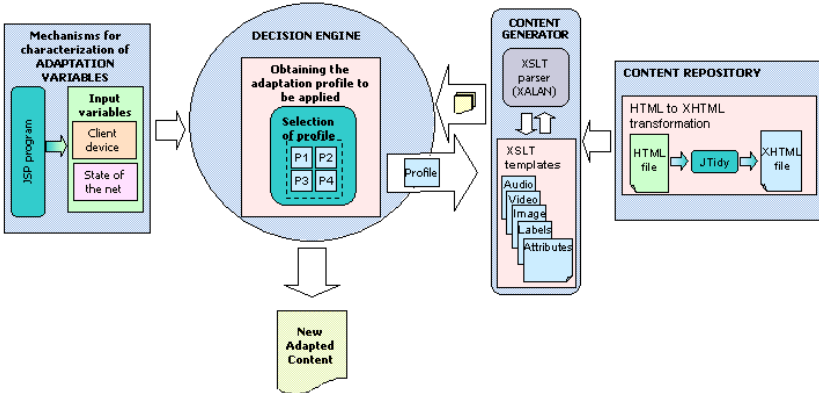


Fig. 3. X-SHAAD structure

³ USD is Spanish acronym of “Teaching Support Unit”

We used XML [18] and XSL for the implementation of X-SHAAD. XSL (eXtensible Stylesheet Language) uses XML notation and is a language for expressing stylesheets. An XSL stylesheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an XML document that uses the formatting vocabulary.

XSL consists of three parts: *XSL Transformations (XSLT)* [19], a language for transforming XML documents; *XML Path Language (XPath)*, an expression language used by XSLT to access or refer to parts of an XML document; and *XSL Formatting Objects (XSL-FO)*, an XML vocabulary for specifying formatting semantics.

2.1 HTML to XHTML Transformation

Currently different browser technologies exist. Some browsers run internet on computers, and some browsers run internet on mobile phones and hand held devices. Each of these handle "bad" marked up HTML documents in a different way. For example, the following HTML could present a problem for them:

```
<html>
<head>
<title> Bad HTML code </title>
<body>
<h1>Bad HTML
</body>
```

Transforming HTML to XHTML solves problems that arise from mal-formed HTML files. XHTML [20] consists of all elements in HTML 4.01 combined with the syntax of XML. HTML was designed to display data and XML was designed to describe data. XML is a markup language where everything has to be marked up correctly, which results in "well-formed" documents

Some of the most important characteristics of XHTML are as follows: XHTML elements must be properly nested, XHTML documents must be well-formed, tag names must be in lowercase, all XHTML elements must be closed and attribute values must be quoted.

In X-SHAAD, for one time only, HTML to XHTML transformation is carried out beforehand through JTidy [21]. It is an open source code program that allows HTML files to be analyzed and to obtain a well formed XHTML. Then, we can apply XSL and its tools to make the transformation.

2.2 Obtaining the Adaptation Profile to Apply

In order to check the operation of the X-SHAAD, the user specifies values of input variables by means of a web interface. The assignment is carried out through a given values list. Table 1 shows the variables considered in this implementation and the set of possible values for each variable. X-SHAAD's Decision Engine infers the adequate profile for each situation and determines a set of adaptation mechanisms will be

applied to the requested web page in the Content Generator. Table 2 shows various adaptation mechanisms to be applied to each media object.

In the current implementation, different static versions of each multimedia object exist in the server to simulate the adaptation mechanisms. When a profile is applied, the directory that contains the elements to send is determined. Table 3 shows two system application examples with the corresponding adaptation mechanisms to be applied.

Table 1. Variables considered and possible value

Variable considered	Possible value
Screen area	High
	Medium (640x480)
	Low (PDA)
Color	Color
	Gray Scale
Video	Yes
	No
Sound	Yes
	No
State of the network	Free
	Semi-busy
	Busy

Table 2. Adaptation mechanisms to be applied to each media object.

Media object	Adaptation mechanism
Image (*)	Image in original size
	Reduced size image to 1/2
	Reduced size image to 1/4
	No image
Video (*)	Original video
	Sequence of 10 images
	Sequence of 5 images
	No video
Sound	Audio file is sent
	Audio file is not sent

(*) Both of these take the values that are indicated in Table 1.

Table 3. Application examples of X-SHAAD operation

Screen Area	Color	Video	Sound	State of Network	Adaptation Mechanism to be applied
High	Color	Yes	Yes	Semi-busy	Image: Reduced image to 1/2 Video: Sequence of 10 images Sound: Audio file is sent
Low	Grayscale	Yes	No	Free	Image: Reduced image to 1/4 Video: Sequence of 5 images Sound: Audio file is not sent

2.3 Obtaining the Adapted HTML

The adaptation of requested HTML files is carried out through the XSLT templates that are applied on obtained XHTML files (section 2.1)

From these XHTML files and using XSL / XSLT, transformations can be implemented in two ways:

- Defining, on a unique XSLT template, all profiles that can be applied to the XHTML document. This template will include as many conditional branches as there are profiles. In order to consider a new profile, a new branch in this template must be introduced.
- Defining individual templates, one for each media object and considering profiles inside these templates. Taking a new profile into account means modifying each

template individually. Considering a new media object implies generating the corresponding XSLT template. This treatment, carried out on individual templates, allows for easier maintenance. It has been selected for our proposal.

Now, to apply an XSLT template to an XHTML file and to introduce parameters, a stylesheet processor should be used, for example SAXON[22] or XALAN[23]. This processor is used to check that the document is well formed and validates against the corresponding DTD and it is in charge of applying the stylesheet XSL, etc.

SAXON, is a processor written in Java with its own parser, although it can use an external one. XALAN, from the Apache XML project, is a free tool written in Java that needs a Java virtual machine (JVM) and an XML parser to work. Xerces is the parser used and the corresponding file (xerces.jar) is included in the distribution.

Since we use an Apache server, we have used XALAN to implement our analyzer.

3 Readdressing of Requested Pages

The X-SHAAD execution is carried out on-line. When receiving the petition, the requested page is analyzed. If the petition is an HTML file, X-SHAAD is executed and the page obtained is adapted to the particular profile.

The module *mod_rewrite* of Apache server is used [24] to do this. It provides virtually all of the functions required to manipulate URLs, and works in most situations. Consequently, *mod_rewrite* can be used to solve all sorts of URL-based problems. We considered two directives of this module:

- *RewriteEngine*. This directive enables or disables the runtime rewriting engine. If it is set to off, then *mod_rewrite* does no runtime processing
- *RewriteRule*. This directive allows us to write the rules that will readdress the requested URLs. The basic syntax of this directive is:

RewriteRule url-pattern url-new

The *url-pattern* is a regular expression that is applied to the current URL. The current URL may not be the URL requested originally, because any number of rules could have already matched and altered it. The *url-new* argument is the string that is substituted for the original URL matched by the *url-pattern*.

The module activation *mod_rewrite* of the Apache is carried out through the file *httpd.conf*. The code below shows part of this file, in which the *mod_rewrite* module is activated, including the rules to apply with the *RewriteRule* directive:

```
<IfModule mod_rewrite.c>
RewriteEngine On
    RewriteRule /adaptador/web/([^\.]+) / ([^\.]+)\. (html|htm)
    /adaptador/jsp/adaptar.jsp?&PATH=$1&XML=$2 [R]
</IfModule>
```

Where:

- [^\.]* : any text that doesn't contain point + /
- PATH* : requested URL address, without the name of the HTML / HTM file
- XML* : name of requested page –HTML / HTM

If the requested page coincides with some of rules specified through the *RewriteRule*, a JSP program is executed that will trigger the processor of style sheets to carry out the adaptation (Xalan). The following command shows the sentence of Xalan execution.

```
java org.apache.xalan.xslt.Process -IN prova.xml -XSL
prova.xsl -OUT prova.out -PARAM perfil 2
```

4 Adaptation Examples

In this section we describe some adaptations. For this experiment, a web site was designed with the purpose of showing how X-SHAAD operates. Table 4 shows the different input variable values considered and the resulting web page.

Table 4. Application examples of X-SHAAD operation

Screen Area	Color	Video	Sound	State of Network	Adaptation Mechanism to be applied
High	Color	No	No	Free	Figure 4a
Medium	Grayscale	Yes	No	Free	Figure 4b
Low	Color	Yes	No	Semi-busy	Figure 4c
Medium	Color	Yes	Yes	Busy	Figure 4d

5 Conclusion

We believe that adaptation can be made in three different “adaptivity dimensions”: *technology*, *user characteristics* and *user interaction*. In this paper, we considered a particular implementation that takes into account the *technological* dimension.

X-SHAAD shows the feasibility of adapting the requested web page according to a series of input variables. The main advantage of the proposed approach is its simplicity. At the same time, it represents a way of implementing SHAAD architecture.

In this way, the final objective is to use SHAAD to obtain an integral system that represents the Adaptive Hypermedia System, and completely takes into account the problem of heterogeneity and independent technologies used to implement it. In this case, XML and Java are a simple way to implement and maintain X-SHAAD.

Further work in the future will include more elaborated profiles for X-SHAAD in order to obtain an integrated support for adapted content delivery. Furthermore, the different “adaptivity dimensions” have so far been considered separately, and the results of these studies must be integrated into one system.



Fig. 4a.



Fig. 4b.



Fig. 4c.



Fig. 4d.

Fig. 4. Web page examples

Acknowledgments. This work has been partially financed by CICYT TEL-99-0976 and Galecia project (UE-SOCRATES-MINERVA).

References

1. Brusilovsky, P.: Adaptive Hypermedia. User Modelling and User-Adapted Interaction 11: 87–110, Kluwer Academic Publishers, Netherlands, 2001.
2. De Bra, P.: Design Issues in Adaptive Web-Site Development. Proceedings of the 2nd Workshop on Adaptive Systems and User Modelling on the WWW, Canada, 1999.
3. Peña, C.I., Marzo, J.L., de la Rosa, J.L.: Intelligent Agents in a Teaching and Learning Environment on the Web. 2nd IEEE International Conference on Advanced Learning Technologies (ICALT2002), Kazan (Russia), September 9–12, 2002, ISBN 0-473-08801-0.
4. Perkins, A.: “UMA Project – Universal Multimedia Access from wired or wireless systems”. NTNU – Department of Telecommunications
5. Campadello, S., Helin, H., Koskimies, O., Misikangas, P., Mäkelä, M., Raatikainen, K.: Using Mobile and Intelligent Agents to Support Nomadic Users. 6th International Conference on Intelligence in Networks (ICIN2000), Bordeaux, France, 2000.

6. Yang, Y., Chen, J., Zhang, H.: Adaptive Delivery of HTML Contents. 9th International World Wide Web Conference – The Web: The Next Generation, Amsterdam, 2000.
7. Ma, W-Y., Bedner, I., Chang, G., Kuchinsky, A., Zhang, H.: “A framework for adaptive content delivery in heterogeneous network environments”. Hewlett-Packard Laboratories.
8. Shaha, N., Dessai, N., Parashar, M. : Multimedia Content Adaptation for QoS Management over Heterogeneous Networks. Proceedings of the International Conference on Internet Computing (IC 2001), Nevada, USA, pp 642–648, Computer Science Research, Education, and Applications (CSREA) Press, June 2001.
9. Abdelzaher, T., Bhatti, N.: “Web Server QoS Management by Adaptive Content Delivery”. In Int. Workshop on Quality of Service, June 1999
10. Abdelzaher, T., Bhatti, N.: “Web Content Adaptation to Improve Server Overload Behavior”. The 8th International World Wide Web Conference, Toronto, Ontario, Canada, 1999
11. Arteaga, C., Fabregat, R.: Integración del aprendizaje individual y del colaborativo en un sistema hipermedia adaptativo. JENUI 2002. Cáceres (España), 10–12 July 2002
12. Cannataro, M., Cuzzocrea, A., Pugliese, A.: A Probabilistic Approach to Model Adaptive Hypermedia System. Proceedings of the International Workshop for Web Dynamics (WebDyn), in conjunction with the 8th International Conference on Database Theory (ICDT 2001), London, UK, January 3, 2001.
13. Cannataro, M., Pugliese, A.: A Flexible Architecture for Adaptive Hypermedia Systems. The Int. Workshop on Intelligent Techniques for Web Personalization (ITWP 2001), in conjunction with International Joint Conference on Artificial Intelligence (IJCAI), Seattle, USA, 2001.
14. Mérida, D., Fabregat, R., Marzo, J.L.: SHAAD: Adaptable, Adaptive and Dynamic Hypermedia System for content delivery. Workshop on Adaptive Systems for Web Based Education (WASWE2002), Málaga (España), 28 May 2002.
15. Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. User Modeling and User-Adapted Interaction, Vol. 6, pp. 87–129, Kluwer academic publishers, 1996.
16. De Bra, P., Calvi, L.: Towards a Generic Adaptive Hypermedia System. Proc. 2nd Workshop on Adaptive Hypertext and Hypermedia, Pittsburgh, pp. 5–11, 1998.
17. De Bra, P., Houben, G.J., Wu., H.: AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. Proceedings of ACM Hypertext’ 99, Darmstadt, pp. 147–156, 1999.
18. Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E.: Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation, 6 October 2000.
<http://www.w3.org/TR/2000/REC-xml-20001006>
19. Clark, J.: XSL Transformations (XSLT) Version 1.0. W3C Recommendation, 1999.
<http://www.w3.org/TR/xslt>
20. Pemberton, S., et al.: XHTML 1.0: The Extensible HyperText Markup Language. W3C Recommendation, 26 January 2000. <http://www.w3.org/TR/2000/REC-xhtml1-20000126>
21. Tidy - <http://www.w3.org/People/Raggett/tidy>
22. Saxon Version 6.3 - <http://users.iclway.co.uk/mhkay/saxon/saxon6.3/index.html#Scope>
23. Xalan-Java Version 2.3.1 - <http://xml.apache.org/xalan-j/index.html>
24. Apache HTTP Server Version 1.3, mod_rewrite Module Information.
http://httpd.apache.org/docs/mod/mod_rewrite.html