XAHM: an XML-based Adaptive Hypermedia Model and its Implementation

Mario Cannataro[1], Andrea Pugliese[1,2]

[1]ISI-CNR, Via P. Bucci, 87036 Rende, Italy
[2]D.E.I.S., University of Calabria, Italy

{cannataro, apugliese}@si.deis.unical.it

Abstract. This paper presents an XML-based Adaptive Hypermedia Model (XAHM) and its modular architecture, for modelling and supporting Adaptive Hypermedia Systems, i.e. hypertext-based multimedia systems that allow user-driven access to information and content personalization. We propose a graph-based layered model for the description of the logical structure of the hypermedia, and XML-based models for the description of *i)* metadata about basic information fragments and *ii)* "neutral" pages to be adapted. Furthermore, we describe a modular architecture, which allows the design of the hypermedia and its run-time support. We introduce a multidimensional approach to model different aspects of the adaptation process, which is based on three different "adaptivity dimensions": user's behaviour (preferences and browsing activity), technology (network and user's terminal) and external environment (time, location, language, socio-political issues, etc.). An Adaptive Hypermedia is modelled with respect to such dimensions, and a view over it corresponds to each potential position of the user in the "adaptation space"; the model supports the adaptation of both contents and link structure of the hypermedia.

1 Introduction

In hypertext-based multimedia systems, the personalization of presentations and contents (i.e. their adaptation to user's requirements and goals) is becoming a major requirement. Application fields where contents personalization can be useful are manifold; they comprise on-line advertising, direct web marketing, electronic commerce, on-line learning and teaching, etc.

The need for adaptation arises from different aspects of the interaction between users and hypermedia systems. Users *classes* to deal with are increasingly heterogeneous due to different interests and goals, worldwide deployment of services, etc. Hypermedia systems should be made accessible from different user's terminals, which can differ not only at the software level (browsing and elaboration capabilities) but also in terms of ergonomic interfaces (scroll buttons, voice commands, etc.). Different kinds of network (e.g. wired or wireless) and other network-related conditions, both static (e.g. available bandwidth) and dynamic (per user bandwidth, latency, error rate, etc), should be considered to obtain a comfortable and useful interaction. Finally, taking into account the spatio-temporal position of the user and other "environmental" conditions can lead to a more effective interaction.

To face some of these problems, in the last years the concepts of user modelling and adaptive graphical user interface have come together in the *Adaptive Hypermedia* (*AH*) research theme [2].

The basic components of Adaptive Hypermedia Systems are the *Application Domain Model*, the *User Model* and the *techniques to adapt presentations* to such model. The Application Domain Model is used to describe the hypermedia basic contents and their organisation to depict more abstract concepts. In addition to traditional models, such as those developed in the Human-Computer Interaction and Database fields, the modelling of AHs requires to consider the different sources that affect the adaptation process. The approach that seems to be the most promising for modelling the Application domain is data-centric, and many researches employ well-known database modelling techniques [1].

The adaptation of the presentation to the User Model can be generally distinguished into *adaptive presentation*, i.e. a manipulation of information fragments, and *adaptive navigation support*, i.e. a manipulation of the links presented to the user [5].

Due to the complexity of user models that usually try to capture user's needs, the adaptation process results in a complex task; it is even more demanding when considering dynamic conditions. To efficiently allow the realisation of user-adaptable presentations, a modular and scalable approach to describe and support the adaptation process should be adopted. In particular:

- The Adaptive Hypermedia model and the Adaptation Process Scheme must allow describing the hypermedia in such a way it is easy to find all the system variables that need to be supported in an adaptive way.
- The User Model has to capture not only the user's explicit behaviour (e.g. browsing activity), but also other implicit aspects regarding his/her environment and its dynamic constraints;
- The architecture must easily and efficiently support the adaptation process. It should be noted that the architecture should be flexible with respect to the kind of adaptivity sources (i.e. it should be easy to add new terminals or new kind of networks to the set of supported ones).

In this paper we present a model for the description of Adaptive Hypermedia, named *XML Adaptive Hypermedia Model* (*XAHM*). XAHM allows describing:

- the logical structure and contents of an Adaptive Hypermedia, underlying the different parts of the hypermedia that should be adapted during the adaptation process (the *what*);
- the logic of the adaptation process, distinguishing adaptation driven by technological constraints and adaptation driven by user needs (the *how*).

The logical structure and the contents of an Adaptive Hypermedia are described along different logical levels; upper (abstract) layers are organised as weighted directed graphs of concepts whereas lower (physical) layers are composed of XML documents describing individual pages of the hypermedia. Such pages include basic multimedia fragments extracted from different data sources and described by XML metadata.

The adaptation scheme is described using a multidimensional approach. Each part of the Adaptive Hypermedia is described along three different "adaptivity dimensions": *user`s behaviour* (preferences and browsing activity); *technology* (network and user`s terminal), *external environment* (time, location, language, socio-political issues, etc.). A view over the Application Domain corresponds to each possible position of the user in the "adaptation space".

The rest of the paper is organised as follows: in Section 2 we describe XAHM and detail the different phases of the construction of an adaptive hypermedia; in Section 3 we propose a technique to classify user on the basis of their behaviour; in Section 4 we show a modular multi-tier architecture for modelling and supporting AHSs, which is entirely based on XAHM; Section 5 outlines conclusions and future work.

2 Adaptive Hypermedia Modelling

This Section presents XAHM, our approach to the modelling of Adaptive Hypermedia. Note that in the rest of the paper the term Application Domain will refer to an Adaptive Hypermedia in a particular Application Domain. We first introduce our proposed multidimensional adaptation scheme, and then we show a graph-based layered model for the Application Domain and a probabilistic interpretation of the hypermedia structure, modelling respectively the logical structure and the "intrinsic properties" of the adaptive hypermedia. XAHM adopts XML essentially because of its flexibility and data-centric orientation: it makes possible to elegantly describe data access and dynamic data composition functions, allowing the use of pre-existing multimedia basic data (e.g. stored in relational databases and/or file systems) and the description of contents in a terminal-independent way.

2.1 Adaptation Space

The goal of AHSs is to adapt contents and presentations to satisfy the user's goals and/or requirements. Some of these goals can be captured analysing the behaviour of the current user or of classes of users; for example, the use of data about user's browsing activity or data mining techniques (e.g. clustering) to discover new knowledge about users can help to reveal *latent* wishes. On the other hand, monitoring user's location, terminal or available network bandwidth can allow satisfying response-time requirements. Many of these conditions can be considered orthogonal; others are correlated.

In the proposed architecture the Application Domain is modelled along three different orthogonal adaptivity dimensions (Fig. 1):

- *User's behaviour* (browsing activity, preferences etc.);
- *External environment* (time-spatial location, language, socio-political issues, status of external web sites, etc.);
- *Technology* (kind of network, bandwidth, Quality of Service, user's terminal, etc).

The position of the user in the adaptation space (Fig.1) is described by a tuple of the form [*B*, *E*, *T*]. Each of the values *B*, *E* and *T* varies over a finite alphabet of symbols. The *B* value, related to the User's Behaviour dimension, captures the *group* the user belongs to (i.e. his/her *stereotype profile*); the *E* and *T* values respectively identify environmental conditions and used technologies. As an example, *B* could vary over {*novice*, *expert*}, *E* over {*summer*, *autumn*, *winter*, *spring*} and *T* over {*HTML-low*, *HTML-high*, *WML*}. A personalised view over the Application Domain corresponds to each point of the adaptation space, e.g. [*expert*, *winter*, *HTML-high*].

With respect to the adaptation of pages, the user's behaviour dimension mainly drives the generation of page contents, while the technology dimension mainly drives the adaptation of *page layout* (*shapes* of presentations, buttons, etc.), *size of transmitted data* (e.g. size of text, image and video resolution, etc.), *kind of transmitted data* (e.g. synthesized speech versus plain text, uncompressed versus compressed data, etc.).

For example, an e-commerce web site could show a class of products that fits the user's expertise (deducted from his/her behaviour), applying a season-dependent price, formatting data with respect to the user's terminal and sizing data on the basis of the network bandwidth.

The AHS monitors the different possible sources that can affect the position of the user in the adaptation space, collecting a set of values, called *User*, *Technology* and *External Variables*. The decision of what variables to consider, made by the author of the hypermedia, depends mainly on the Application Domain. The current position of the user [*B*, *E*, *T*] is obtained by means of a mapping function; for example, let us consider *n* Technology Variables, each of which having an associated domain V_i (i=1, ..., n) consisting of a finite alphabet of labels. A simple mapping function

$$f: V_1 \times V_2 \times ... \times V_n \rightarrow T$$

(where *T* can have $|V_1|*|V_2|*...*|V_n|$ values at maximum) could identify the position of the user along the *T* axis. The mapping functions for the Technology and Environment Variables are straightforward, while the mapping from the User Variables to the User Profile is carried out by an algorithm that makes use of a probabilistic interpretation of the link structure of the hypermedia (see Section 3).

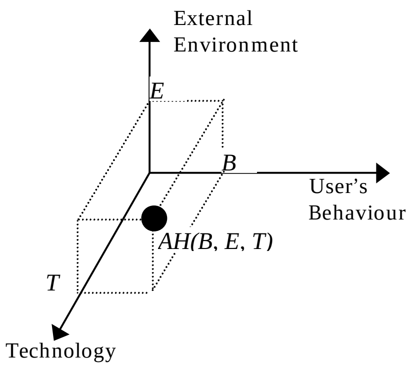

Fig. 1. Adaptation space and adaptivity dimensions

The Application Domain model remains abstract with respect to the alphabets of labels of dimension variables' domains; this feature is significant for the extensibility of the model, i.e. when an author needs to make the dimension variables feasible for a particular domain. For example, referring to the technology dimension, the author could freely split the point *WML* in two disctint points *WML-high* and *WML-low*; in the next Section, it will be shown how such extensions are reflected in new views over the Application Domain, obtained considering further parameters and modelling presentations subsequently.

2.2 A Layered Model for the Logical Structure of the Hypermedia

The proposed Application Domain Model uses a layered data model for describing the logical structure of the hypermedia. Here we use the notion of *directed* graph (a graph $[E, V]$ with two functions *init*: $E \rightarrow V$ and *ter*: $E \rightarrow V$ assigning to every edge an initial and a terminal vertex [11]) to capture the organisation of some layers of the model.

The layered data model extends the Adaptive Data Model described in [9]; it comprises the following abstract levels:

0. *Information Fragments* (*IF*) or *atomic concepts*, like texts, sounds, images, videos, etc. at the lowest level. The information fragments are stored in databases and/or file systems, both local and remote. In fact, very often these are pre-existing data and it could not be suitable to convert their formats. Data can be structured, semi-structured or unstructured and can be provided by different sources (e.g. external or local databases, XML and HTML documents, texts, files and so on). The IFs are described by metadata represented by XML documents.
1. *Presentation Descriptions* (*PD*), XML documents constrained by a fixed DTD, which capture the so-called *Page Concepts* [10]. They comprise multimedia contents, presentation layout and format, access rights to data (where applicable, e.g. when a presentation unit refers to a database table) etc. Page elements are parameterised with respect to the three adaptivity dimensions, so they can be associated to a portion of the adaptation space. Included basic multimedia fragments are referenced by means of the XML metadata describing them. The *final pages* composed of actual fragments, also called *Presentation Units* (*PU*), are dynamically generated at run time in a target language (XML, HTML, WML, VoiceXML, synthesised speech etc.) and delivered.
2. *Elementary Abstract Concepts* (EAC) representing larger units of information. An Elementary Abstract Concept is composed by one or more Presentation Descriptions organised in a digraph, whose links are annotated by a weight. Arcs represent relationships between elementary concepts or navigation requirements (e.g. a sequence of elementary concepts to be learned

before learning an abstract concept), while weights represent their relevance with respect to each other. The linking structure of EACs is differentiated with respect to the user's behaviour adaptivity dimension; an EAC is represented as a digraph as there can be more arcs between the same two nodes, each of which associated to a different user's profile.

3. *Application Domain*. Finally, an Application Domain is composed by a set of Elementary Abstract Concepts organised in a digraph. Arcs represent relationships between EACs; they are differentiated with respect to the user's behaviour adaptivity dimension, but a "null" weight is associated to them as they are used only for describing relationships, and the user can not perform any choice on them.

The Application Domain and the EACs are differentiated with respect to user's profile only, because they need to be directly described by the author and it could not be suitable to build and maintain a large variety of weighted digraphs. However, future extensions of the system could support the personalization of the link structure of the hypermedia with respect to Technology Variables, so allowing "lighter" (i.e. with shorter paths) versions of the hypermedia, to be browsed in a more agile way.

It should be noted that the adaptation of hypermedia contents, by means of transformation from Presentation Descriptions to final pages, comprises both adaptive presentation and adaptive navigation support; both are obtained instantiating each Presentation Description with respect to the user's position in the adaptation space.

2.3 Probabilistic Interpretation of the Adaptive Hypermedia Schema

In the layered model presented in Section 2.2, we introduced a weight in the graphs representing Elementary Abstract Concepts to express links' relevance with respect to each other. In this Section, we propose a probabilistic interpretation of arcs' weight, which is used also for the classification of users on the basis of their behaviour (see Section 3).

Here, we consider the overall Application Domain as a weighted digraph of Presentation Descriptions, i.e. a "plain" version of the AD, obtained from the layered one. Formally, an AD with M different profiles is a set N of Presentation Descriptions where the generic description $i \in N$ contains, for each profile $k=1, \ldots, M$, a set of weighted outgoing links (i, j, k) where j is the destination node. It can be mapped in a weighted digraph $G = (N, E)$ where each node corresponds to a description and each directed arc to an outgoing link; the digraph G can also be referred to as the set of the weighted graphs G_k, $k=1,\ldots,M$, obtained extracting from G the nodes and arcs corresponding to each profile. Each G_k is named *Logical Navigation Graph*.

Our probabilistic interpretation assumes that the weight $W_k(i,j)$ of the arc (i, j, k) is the probability that a user belonging to the profile k follows the link to the j node having already reached the i node:

$$W_k(i, j) = P(j|k,i)$$

$P(i|k,i)$ is considered to be always zero, as it is impossible a link from a node to itself. For each node i, the sum of the weights of outgoing arcs, for each profile, is always one.

We define a path S in G as the ordered set of arcs

$$S = \{ (S_j, S_{j+1}, profile_j) \mid (S_j, S_{j+1}, profile_j) \in E, j=0, \ldots, l-1 \},$$

where $profile_j \in \{1, \ldots, M\}$ represents the group the user belongs to when he/she reaches the node S_j. It should be noted paths involving different Logical Navigation Graphs are allowed such paths refer to the case of a user moved within different profiles during his/her browsing activity.

The probability that a user belonging to the profile k follows the S path is

$$P_S^k = \prod_{j=0..l-1} W_k(S_j, S_{j+1})$$

so P_S^k is the product of the probabilities associated to the arcs belonging to the *S* path, while the "shortest" path $\tilde{S}_{ij}^k$ between two nodes *i* and *j* for a given profile *k* is the path with the maximum joint probability given as

$$\tilde{P}_{ij}^k = \max_{S_{ij}^k} (P_{S_{ij}^k}^k)$$

where S_{ij}^k is the generic path between the nodes *i* and *j* through arcs belonging to the profile *k*.

2.3.1 Static Properties of Hypermedia Structure

In our model, we consider some *static* (*intrinsic*) *properties* of the hypermedia structure and construct three discrete probability density functions (*PDF*):

- $\mu(k)$, for each profile *k*, proportional to the mean value of the probability of the "shortest" paths in G_k; high values of this PDF indicate the existence of highly natural paths in the hypermedia.
- $p(k)$, for each profile *k*, proportional to the mean value of the length of the "shortest" paths in G_k; high values of this term mean longer natural paths in the hypermedia, which could be an advantage in the overall personalization process.
- $n(k)$, for each profile *k*, proportional to the number of nodes belonging to the profile.

It should be noted that these values can change over time: the hypermedia structure can dynamically be updated (adding or removing nodes, arcs or their weight) on the basis of semi-automatic observation of the behaviour of many users or on the basis of an increased knowledge of the Application Domain by the author.

A weighted medium, expressing the "intrinsic relevance" of the profiles is computed:

$$s(k) = \frac{\beta_0 \mu(k) + \beta_1 n(k) + \beta_2 p(k)}{\beta_0 + \beta_1 + \beta_2}$$

where the values of $\mu(k)$ and $n(k)$ should be traded-off as a profile with few nodes could have few paths with higher probabilities. An high value of each of the terms in $s(k)$ expresses a high relevance with respect to the profile *k*, so $\beta_i>0$.

3 User Classification

The proposed probabilistic interpretation of the hypermedia structure is used to characterise "latent" properties of the user's behaviour, which can be captured by tracking his/her browsing activity. Such properties, related to the user's behaviour adaptivity dimension, are expressed by means of an association of the user to a stereotype model. In this Section we describe our approach to such classification task.

The proposed system builds a discrete probability density function $A(k)$, with k=1, …, M, measuring the "belonging probability" of the user to each group (i.e. how much each profile fits him/her). While the user browses, the system updates $A(k)$ and the user's profile is changed accordingly. In other words, on the basis of the user's behaviour, the system dynamically attempts to assign the user to the "best" profile.

Browsing starts from the presentation unit associated to a starting node. If the user is already registered, the last $A(k)$ is set as current. Otherwise, he/she is assigned to a generic profile, or to one calculated on the basis of a questionnaire (see [4] for an interesting way to interpret results in a probabilistic way); the initial value of $A(k)$ is called $A_0(k)$. When the user visiting the node R_{r-1} requests to follow a link, the system computes the new PDF $A'(k)$, on the basis of the User Behaviour Variables and of $s(k)$ (see Section 2.3.1), then it decides the (new) profile to be assigned to the user. To avoid continuous profile changing it is possible to keep a profile for a given duration (i.e. the number of traversed links), evaluating the $A'(k)$ distribution at fixed intervals.

The user's behaviour is stored as a set of User Behaviour Variables:

- The current profile, k_c;
- The current discrete PDF $A(k)$, $k=1,\ldots,M$, measuring the user's "belonging probability" to each profile;
- The recently followed path $R = \{R_1, \ldots, R_{r-1}, R_r\}$, which contains the last visited nodes, where R_{r-1} is the current node and R_r is the next node. The last arc (R_{r-1}, R_r, k_c) is the outgoing link chosen by the user;
- The time spent on recent nodes, $t(R_1)$, ..., $t(R_{r-1})$.

On this basis, the system constructs three PDFs:

- $c(k)$, proportional for each profile k to the probability P_R^k of having followed the R path through arcs belonging to the profile k; a high value of P_R^k indicates that the visited nodes in R are relevant for the profile k as the actual path is "natural" for the profile k.
- $r(k)$, proportional for each profile k to the reachability $\tilde{P}_{R_1,R_r}^k$ of the next node R_r from the first node R_1, through arcs belonging to the profile k. This term takes into account the way the user *could have reached* the next node R_r; in fact, a high value means that there exists a very "natural" way to reach it through links of the profile k.
- $t(k)$, proportional for each profile k to the distribution $D^t[k]$ of the visited nodes from R_1 to R_{r-1}, weighted with the time spent on each of them, with respect to the profile k. For example, let $\{ n_1, n_2, n_3 \}$ be the recently visited nodes and $\{ t_1, t_2, t_3 \}$ the time units spent on each of them: if node n_1 belongs to profiles k_1 and k_2, node n_2 belongs to k_2 and k_3 and node n_3 belongs to k_1 and k_4, the distribution is evaluated as $D^t[k] = [(k_1, t_1+t_3), (k_2, t_1+t_2), (k_3, t_2), (k_4, t_3)]$. $D^t[k]$ shows how the time spent on visited nodes is distributed with respect to profiles, and is obviously an indicator of the interest the user has shown with respect to them. The visiting times should be accurate; an interesting approach for an accurate computation is proposed in [12].

Temporary deviations that do not move the user's interests can be taken into account trading off the effects of $c(k)$ and $r(k)$ on $A(k)$. The former takes into account the actual path so it aims to move towards the profile corresponding to recent preferences, whereas the latter aims to disregard recent (local) choices, as the "shortest" paths not necessarily consider the visited nodes between R_1 and R_r.

Only the most recently followed r-1 links (r nodes) are considered, to avoid an "infinite memory" effect. In fact, considering R from the initial node, the probability P_R^k of having followed R in the profile k is zero if the user visits just one node not belonging to the profile k (obviously we consider $W_k(i, j) = 0$ if $(i, j, k) \notin E$).

Finally, a weighted medium expressing the "dynamic relevance" of the profiles is computed:

$$d(k) = \frac{\alpha_0 c(k) + \alpha_1 r(k) + \alpha_2 t(k)}{\alpha_0 + \alpha_1 + \alpha_2}$$

An high value of each of the terms in $d(k)$ expresses a high relevance with respect to the profile k, so $\alpha_i > 0$.

The algorithm that computes the new PDF $A'(k)$ takes as input *(i)* the discrete PDFs $A(k)$, $A_0(k)$ and $s(k)$, *(ii)* the recently followed path $R = \{R_1, \ldots, R_{r-1}, R_r\}$ and *(iii)* the time spent on recently visited nodes, $t(R_1)$, ..., $t(R_{r-1})$. It computes the new $d(k)$ and applies the formula

$$A'(k) = \frac{\gamma_0 A_0(k) + \gamma_1 A(k) + \gamma_2 d(k) + \Delta\gamma_3 s(k)}{\gamma_0 + \gamma_1 + \gamma_2 + \Delta\gamma_3}, \text{ where } \Delta = \begin{cases} 1, \text{if } s(k) \text{ has changed} \\ 0, \quad \text{otherwise} \end{cases}$$

The algorithm combines the user's dynamic behaviour, synthesised in the term $d(k)$, with the structural properties of the hypermedia scheme, mainly depending on its topology, synthesised in the term $s(k)$. The new $A'(k)$ is computed as a weighted medium of four terms, also considering the initial user's choices and the story of the interaction. An high value of each of the terms in $A'(k)$ expresses a high relevance with respect to the profile k, so $\gamma_i > 0$. The new profile could be chosen making a random extraction over the $A'(k)$ distribution or referring the highest $A'(k)$ value.

4 System Architecture

In this Section we present the architecture for the construction and the run-time support of XAHM-based systems. After a description of our use of XML and XML-related technologies, we show the run-time support of the system and a set of authoring tools for the design and test of the AHs.

4.1 XML Metadata and Presentation Descriptions

In XAHM both pages and metadata are described by using XML. Each data source is "wrapped" by an XML meta-description, whereas each Presentation Description is a XML document obeying a well-defined structure, described in the following. The use of "pure" XML instead of more widespread formalisms for metadata, such as RDF and RDF Schema, allows a simpler and more direct support to the proposed multidimensional approach.

The use of metadata is a key aspect in order to accomplish the multidimensional adaptation task; for example, an image could be represented using different levels of detail, formats or points of view (shots), whereas a text could be organised as a hierarchy of fragments, represented using different languages, or an XML document could be differentiated along different "detail levels" [6]. These different versions of the same data could be associated to different points of the multidimensional adaptation space. Furthermore, by means of meta-descriptions, data fragments of the same kind can be treated in an integrated way, regardless of their actual sources: in the construction of pages the author refers to metadata, thus avoiding too low-level access to fragments.

A number of *Document Type Definitions* [13] for the XML meta-descriptions have been designed. They comprise descriptions of:

- *text*, hierarchically organized;
- object-relational *database tables*;
- *queries versus object-relational databases*;
- *queries versus XML data*, expressed in *XQuery* [13];
- *images* and *video sequences*;
- *XML documents*; *HTML documents*.

As said before, in our system the Presentation Descriptions are XML documents whose key parts are the *content, fragment* and *embedded-code* elements. The *content* element is used to include text in the page. The *fragment* element is useful for including basic multimedia fragments referenced by their aliases. Finally, the *embedded-code* element increases flexibility allowing the insertion of terminal-dependent code (e.g. WML, HTML) in the page (obviously, wrapped by an XML `CDATA` section).

Each part of the PD is organised as a sequence of elements; each of them can be associated to a portion of the adaptation space by means of the *dimension-parameters*, i.e. dimension variables interpreted here as parameters. The dimension-parameters can be any XML `NMTOKENS` set and, as seen in Section 2.1, the author is allowed to decide the alphabet of labels regarding such parameters. Before storing the Presentation Descriptions, the system actually adds to them some *XSP* tags [3], containing portions of high-level code to be executed at run-time for instantiating them.

4.2 The Run-Time System

The run-time system supporting XAHM has a *three-tier* architecture (Fig. 2), comprising the *Presentation*, the *Application* and the *Data Layers.*

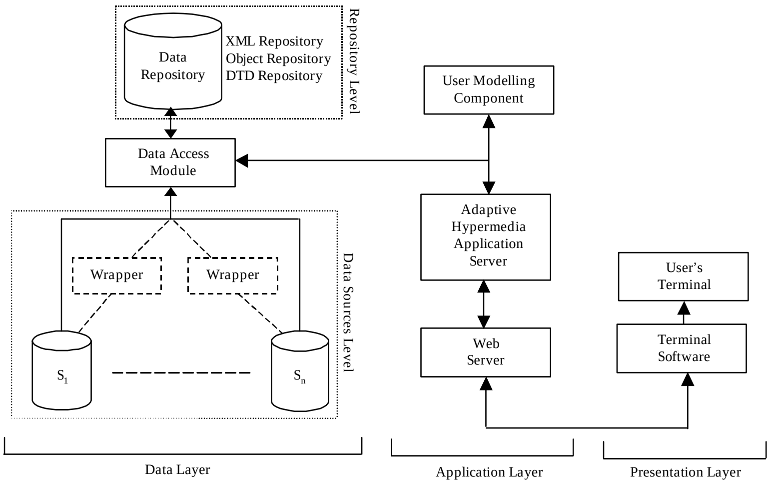

Fig. 2. Run-time system architecture

The Presentation (User) Layer receives final pages to be presented and eventually scripts or applets to be executed; these scripts are useful for detecting the *user context*, e.g. local time, physical location, available bandwidth or the time spent on pages. The user's terminal and the terminal software (operating system, browser etc.) are typically communicated by the terminal User Agent (browser).

The Application Layer is the core of the system: it collects the user behaviour and characteristics and implements the adaptation process. It comprises two main modules: the *Adaptive Hypermedia Application Server* (*AHAS*) and the *User Modelling Component* (*UMC*) [4]; they run together with a *Web Server*. The UMC maintains the most recent actions of the user and executes the algorithm for the evaluation of the user's profile. After a user has selected the next page and the system has determined his/her user's position in the Adaptation Space, the AHAS executes the following steps:

1. extracts from the XML repository the Presentation Description to be transformed and executes the application logic contained in it. The logic comprises the extraction and composition of basic data fragments from the data sources on the basis of the known user position;

2. extracts the XSL stylesheet from the XML repository needed to adapt the page layout to the user's terminal and applies it to the XSP document;
3. returns the final page to the Web Server.

Finally, the main goal of the Data Layer is to store persistent data and to offer efficient access primitives. It comprises the *Data Sources Level*, the *Repository Level* and a *Data Access Module*. The Data Sources Level is an abstraction of the different kinds of data sources used to build the hypermedia. Each data source S_i is also accessed by a *Wrapper* software component, which generates in a semi-automatic way the XML metadata describing the data fragments stored in S_i.

The Repository Level is a common repository storing data provided by the Data Source Level or produced by the author. It stores:

- XML documents into a *XML Repository*; these documents include source Presentation Descriptions (as XML documents), generated XSP Presentation Descriptions, XSL stylesheets and XML metadata.
- persistent objects into an *Object Repository*; the objects represent Logical Navigation Graphs and data about registered users.
- the XAHM DTDs used to validate XML documents.

Finally, the Data Access Module implements an abstract interface for accessing the Data Sources and the Repository levels.

4.3 The Java Adaptive Hypermedia Suite (JAHS)

According to the described architecture, we have designed and implemented a set of Java-based tools allowing the design, the simulation and the validation of an Adaptive Hypermedia based on XAHM, through an iterative and interactive process. Using a *RAD* (*Rapid Application Development*) approach, the author first defines the overall structure of the hypermedia, then simulates the behaviour of the system on the basis of different classes of users and adjusts the hypermedia structure accordingly. Then, the author can complete the hypermedia construction providing the contents of the PDs.

4.3.1 Multidimensional Adaptive Hypermedia Authoring

In the construction of an Adaptive Hypermedia the following main phases can be identified, almost directly related to the layered model of Section 2.2.

High-Level Structure Definition

The high-level structure of an adaptive hypermedia is modelled by means of the first two layers of the graph-based model described in Section 2.2. The author first defines the set of stereotype user profiles representing users' groups. Subsequently, he/she describes the overall Application Domain as a digraph of EACs using the *Hypermedia Modeller*, a tool that allows designing EACs in a visual way. Finally, the author describes each EAC, specifying sets of PDs, differentiating links with respect to user profiles and adding to them the probabilistic weights. Notice that in this phase it is not necessary to specify the PD's content, but only their link structure. The Hypermedia Modeller provides some hints about typical graph structures and offers a set of utilities regarding the overall probabilistic structure of the hypermedia (shortest paths, minimum spanning tree, etc.).

The *Graph Object Validator*, which validates the graph descriptions of the hypermedia (e.g. with respect to coherence of probabilities, congruence with the links contained in the Presentation Descriptions, etc.), generates the persistent objects containing the weighted digraphs and stores them. The use of persistent representation allows reusing parts of the hypermedia; thus, after having been validated and stored, (part of) objects can be imported by the Hypermedia Modeller to design new EACs.

Semi-Automatic Metadata Creation
Since basic multimedia information fragments are always accessed by means of metadata associated to them, a fundamental step is concerned with the creation of such metadata. The *Fragments Browser/Composer* allows browsing the information fragments provided by the Data Sources Level, using some Wrapper software components, and extracting some explicit metadata. Moreover, the author can add information on the basis of his/her domain knowledge. As an example, the Fragments Browser/Composer is able to connect to local or remote DBMS and automatically extract the structure of relational or object-oriented tables; or it can explore local or remote file systems and extract metadata about stored files of known types. The author of the hypermedia is allowed to integrate such metadata (e.g. with human-readable explanations) or to create new ones (e.g. descriptions of typical queries).

Presentation Descriptions Construction
The last (and typically longest) phase of the AH design is the construction of the Presentation Descriptions. Here, the author composes basic information fragments, referencing their metadata, and associates them to specific portions of the adaptation space by means of parameters regarding the adaptivity dimensions. This phase is performed by using a *PD Editor*, which allows editing XML Presentation Descriptions in the form of pure text or graphically (as pure trees, or in a "visual" way). It is possible to create new documents and to edit pre-existing ones; the PD editor also allows a "preview" of the final pages. A *PD Transformer* performs a validating parsing of XML Presentation Descriptions with respect to the DTDs, adds to them the XSP tags and stores them into the repository.

Notice that the author can use the top-down approach described above, or choose a bottom-up approach, starting from the definition of the PDs. We chose to adopt a *procedural* approach in the definition of the high-level structure of the Application Domain since it allows a simpler implementation with respect to *declarative* modelling approaches.

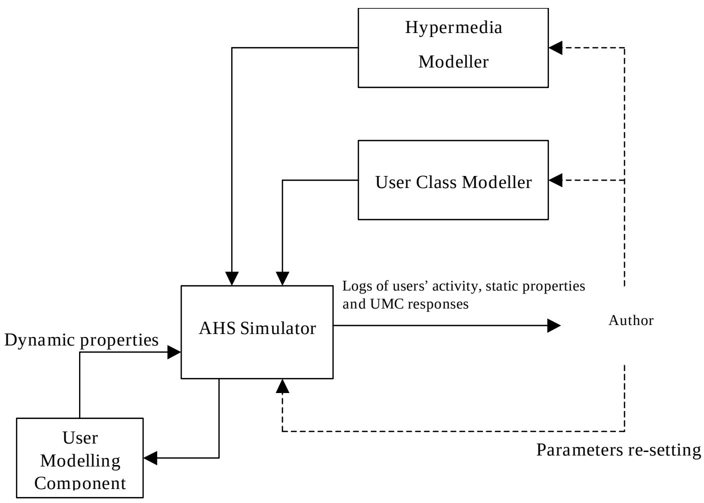

Fig. 3. The Simulation Tool

4.3.2 Simulation and Validation of the Adaptive Hypermedia

Generally, it is fundamental for an author to validate the high-level link structure of the hypermedia with respect to the mechanisms that drive the profile assignment decision. This is especially true in the proposed system, where links are weighted by probabilities. Therefore, the system provides a *Simulation Tool* (Fig.3) that permits the author of the hypermedia to:

1. Analyse the intrinsic properties of the hypermedia (see Section 2.3.1), calculated from its structure.
2. Define a set of *Classes* of typical users whose behaviour needs to be simulated, by means of the *User Class Modeller*, to validate the response of the system. Many different User Masks can be assigned to each class, so the behaviour of a user can change during the same interaction with the system; clearly, behaviours modelled by means of User Classes comprise random visiting times or choice of arcs.
3. Run the simulation by means of the *AHS Simulator*, that is a multithreaded machine that generates requests of a number of users to the AHAS, and presents the resulting logs in a graphical way.
4. Analyse the profile assignment decision (i.e. the response of the UMC) with respect to the User Classes.
5. Eventually (e.g. in the case of many *oscillations* of resulting PDFs), *(i)* tune the parameters used in the algorithm, as the length of the sliding temporal window or the values of the parameters used to weight the PDFs, or *(ii)* adjust the hypermedia structure.

5 Concluding Remarks and Future Work

In this paper we presented XAHM, an XML-based model for Adaptive Hypermedia Systems. XAHM models an Application Domain (i.e. the hypermedia) considering a three-dimensional adaptation space, comprising the user's behaviour, technology, and external environment dimensions. The adaptation process is performed finding the proper position of the user in the adaptation space, and applying to "neutral" XML pages some constraints bound to that position.

We believe that the main contributions of this paper are:

- A new model to describe Adaptive Hypermedia allowing a flexible and effective support of the adaptation process.
- A probabilistic model of the user's behaviour, and a classification algorithm that attempts to accomplish the profiling task in an effective and non-invasive way.
- A scalable and modular architecture for the design and the run-time support of Adaptive Hypermedia System.

Future work will concern the completion of the implementation and the test of the UMC response with respect to some canonical hypermedia structure and typical users' behaviour (in the final version of the paper we will show some experimental results). Moreover, we will introduce Data Mining techniques, to let the author examine the actual behaviour of a number of users and fine-tune the profiles' probabilities accordingly.

References

1.Abiteboul, S., Amann, B., Cluet, S., Eyal, A., Mignet, L., Milo, T., "Active views for electronic commerce", in *Proceedings of the 25th VLDB Conference*, 1999.

2.Adaptive Hypertext and Hypermedia Home Page, http://wwwis.win.tue.nl/ah/

3.Apache Software Foundation, "The Apache XML Project", http://xml.apache.org.

4.Ardissono, L., Goy, A., "Tailoring the Interaction With Users in Web Stores", in *User Modeling and User-Adapted Interaction*, 10(4), Kluwer Academic Publishers, 2000.

5.Brusilovsky, P., "Methods and techniques of adaptive hypermedia", in *User Modeling and User Adapted Interaction*, v.6, n.2-3, 1996.

6.Cannataro M., Carelli G., Pugliese A., Saccà, D., "Semantic lossy compression of XML data", *8th Int. Workshop on Knowledge Representation meets Databases* (KRDB-2001) in conjunction with *VLDB 2001*.

7. Cannataro, M., Cuzzocrea, A., Pugliese, A., “A probabilistic approach to model adaptive hypermedia systems”, in *Proceedings of the International Workshop on Web Dynamics*, 2001.
8. Ceri, S., Fraternali, P., Bongio, A., “Web Modelling Language (WebML): a modelling language for designing web sites”, *WWW9 Conference*, 2000.
9. De Bra, P., Aerts, A., Houben, G.J., Wu, H., “Making General-Purpose Adaptive Hypermedia Work”, *Proceedings of the WebNet Conference*, pp. 117-123, 2000.
10. De Bra, P., Brusilovsky, P., Houben, G.J., “Adaptive Hypermedia: from systems to framework”, *ACM Computing Surveys, Symposium Issue on Hypertext and Hypermedia*, 1999.
11. Diestel, R., *Graph Theory*, Springer-Verlag, New York, 2000.
12. Murtagh, F., Tao, F., “Towards knowledge discovery from WWW log data”, in *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2000.
13. World Wide Web Consortium, http://www.w3.org.