# A Flexible Architecture for Adaptive Hypermedia Systems

Mario Cannataro and Andrea Pugliese
*ISI-CNR, Via P. Bucci, 41/c - c/o DEIS-University of Calabria, 87036 Rende, Italy*
*{cannataro, apugliese}@si.deis.unical.it*

## Abstract

This paper presents an XML-based flexible and modular architecture for the modelling and the run-time support of web-based Adaptive Hypermedia Systems supporting multi-channel access. The proposed *XML Adaptive Hypermedia Model* (XAHM) and the supporting architecture allow the hypermedia adaptation along three different "adaptivity dimensions": *user`s behaviour*; *technology, external environment*. A view over the Application Domain corresponds to each possible position of the user in the "adaptation space". The user's behaviour is modelled using a probabilistic approach that dynamically constructs a discrete probability density function. Using that distribution the system attempts to assign the user to the "best" profile that fits his/her expectations. Moreover, the delivered presentation units are composed, formatted and presented according to network and terminal constraints, and external environment conditions.

## 1. Introduction

The linking mechanism of hypermedia offers users a large amount of navigational freedom so that it becomes necessary to offer support during navigation. The personalisation of web presentations and contents, i.e. their adaptation to user's requirements and goals, is becoming a major requirement of modern web-based systems. Different application fields where contents personalisation can be useful are on-line advertising, direct web-marketing, electronic commerce, on-line learning and teaching, etc.

Main aspects that such systems should take into account are:

- the different classes of users that will use the system; they are becoming more and more heterogeneous due to different interests and goals, world-wide deployment of services, social conditions, etc.;
- the kind of user terminals and network; user terminals can differ not only at the software level (browsing and elaboration capabilities), but also in terms of ergonomic interfaces (scroll buttons, voice commands, etc.); network can differ with respect to their type (wired, wireless) and dynamic properties (per user bandwidth, latency, error rate, etc.;
- the external environment conditions, such as socio-economical issues, the connection time and user location etc.

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. Some recent Adaptive Hypermedia Systems (AHS) are outlined in [10].

Basic components of Adaptive Hypermedia Systems are the Application Domain Model, the User Model and the techniques to adapt presentations with respect to the User Model. The Application Domain Model is used to describe the hypermedia basic contents and their organisation to depict more abstract concepts. The most promising approach in modelling the Application domain is data-centric, and many recent researches employ well known database modelling techniques [1, 13].

The User Model attempts to describe the user's characteristics and preferences and his/her expectations in the browsing of hypermedia. User Models (*profiles*) can be distinguished in *overlay models*, which describe a set of user's characteristics, typically represented by a set of *attribute-value* pairs, and *stereotype models* which indicate the user's belonging to a group [8]. Obviously, the User Model comprises all the user's characteristics, not only "behavioural" ones.

The adaptation of the Application Domain 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. More recently, the capability to deliver a given content to different kind of terminals, using wired or wireless networks, i.e. the support of multi-channel accessible web systems, is becoming an important requirement of AHS.

Due to the complexity of user models that usually capture explicit user needs, the adaptation process results in a complex task and it is more demanding when considering more dynamic conditions, such as the available network bandwidth, the time/location of access, and other implicit user needs.

To efficiently allow the realisation of user-adaptable contents and presentation, a modular and flexible approach to describe and support the adaptation process should be adopted and a clear separation between basic multimedia contents, Domain Model and User Model should be achieved. Whereas the conceptual model offered by standard database technologies can be easily employed to describe basic fragments of multimedia contents (such as images, audio and video, texts, sounds, etc.), the Domain Model and the User Model need more sophisticated, dynamic and adaptable conceptual models. The Domain Model should allow to easily describe, eventually using graphical tools, the application domain, i.e. its concepts and relationships, whereas the User Model should capture the behaviour and the expectations of users. Moreover, these conceptual models should be supported by flexible and modular architectures providing efficient access to basic data fragments, their dynamic selection, extraction and composition.

Following those requirements, in this paper we present an XML-based flexible and modular architecture for the modelling and the run-time support of web-based Adaptive Hypermedia Systems supporting multi-channel access. The proposed Application Domain model, named *XML Adaptive Hypermedia Model* (XAHM), and the supporting architecture allow the hypermedia adaptation 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 such "adaptation space". To present the proper contents to the user, a set of values, called *User*, *Technological* and *External Variables,* are collected monitoring respectively user's activity, technological constraints and environmental conditions.

Using collected user's behaviour values, a stereotype User Model is constructed. The stereotype model combines such values with intrinsic properties of the hypermedia structure, to construct a discrete probability density function measuring how much each profile fits a user; i.e. the probability that a user belongs to each profile is updated as long as the browsing goes on. Using that distribution the system attempts to assign the user to the "best" profile that fits his/her expectations. Moreover, the delivered presentation units are composed, formatted and presented according to network and terminal constraints, and external environment conditions.

The proposed model is being used as a basis for the design and development of an Adaptive Hypermedia Architecture that uses the above-mentioned probabilistic approach for the adaptation process.

The rest of the paper is organised as follows. Section 2 presents the XAHM model. Section 3 describes the probabilistic hypermedia scheme, the probabilistic user model and a technique to assign users to profiles. Section 4 depicts the system architecture and Section 5 outlines conclusions and future work.

## 2. XAHM: an XML-based Adaptive Hypermedia Model

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, whereas different goals are latent or are considered to be "obvious". For example, the use of data mining techniques to discover new knowledge about a user (clustering, classification, etc.) can help to reveal latent wishes, whereas monitoring location, user's terminal or available network bandwidth can allow satisfying response-time requirements. Many of these conditions can be considered orthogonal, i.e. non-correlated; other ones are correlated.

This Section presents our approach to the modelling of AHs. After a description of the proposed adaptation scheme, we show *i)* a graph-based layered model for the Application Domain and *ii)* the XML-based models used to describe metadata about basic information fragments and "neutral" pages to be transformed and delivered to the user.

### 2.1. Adaptation Scheme

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, external web sites content, etc.);
- Technology (kind of network, bandwidth, Quality of Service, user's terminal, etc).

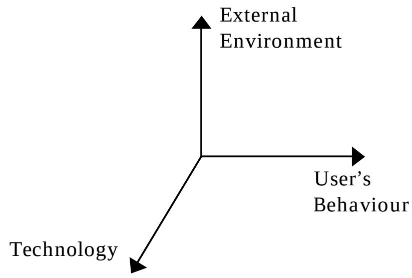


**Fig. 1.** Adaptivity dimensions

The position of the user in the adaptation space of 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 (e.g. natural numbers, without loss of generality). 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 used technologies and environment location.

The Adaptive Hypermedia is a sub-space $AH(B, E, T)$ of the overall adaptation space, specified by the author that provides its description for some particular portions of the space. In turn, a personalised view over the Application Domain corresponds to a sub-space of the Adaptive Hypermedia; the definition of such sub-spaces, also given by the author, can outline some overlap among them.

The user's behaviour and external environment dimensions mainly drive the generation of page content, while the technology dimension mainly drives the adaptation of page layout. For example, an e-commerce web site could show a class of products that fits the user's needs (deducted from his/her behaviour), applying a time-dependent price (e.g. night or day), formatting data w.r.t. the user terminal and sizing data w.r.t. 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*, *Technological* 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 from the corresponding variables to mono-dimensional values. For example, let us consider $n$ Technological Variables, each of which having an associated domain $V_i$ ($i$=1, ..., $n$) consisting of a finite alphabet of labels; there will exist a mapping function

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

where $T$ can have $|V_1|*|V_2|*...*|V_n|$ possible values. The mapping functions for the Technology and Environment Variables can be simple, while the mapping from the User Variables to the User Profile is carried out by an algorithm, described in detail in Section 3.

The Application Domain model remains abstract w.r.t. the alphabets of labels composing the domains of the dimension variables; this feature is significant for the flexibility of the model, i.e. when an author needs to make the dimension variables feasible for a particular domain (see Sec.2.3).

### 2.2. The Layered Data Model

The layered data model extends the Adaptive Data Model described in [15]; it comprises the following abstract levels:

0. *Information Fragments* (IF) like texts, sounds, images, videos, and their metadata, represented as XML documents, at the lowest level. The information fragments are stored in databases, files or everywhere in the Web. 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). They are described by metadata represented by XML documents stored in a XML repository (a relational database or a file system). The use of metadata is the key aspect to support adaptation along different dimensions (see Sec.2.3).
1. *Presentation Descriptions* (*PD*) composed by XML documents stored in the XML repository. A Presentation Description describes what information fragments can occur in a presentation unit, and how these data can be selected on the basis of different *parameters*, such as user's profile, technological conditions and external variables.
   - At run time the *Presentation Units* (*PU* or *pages*), i.e. the pages composed of fragments that are presented to the user, are obtained from Presentation Descriptions. PDs are instantiated with respect to user's profile and external variables, and final pages are dynamically generated in a target language (XML, HTML, WML etc) depending on technological constraints.
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 weighted digraph. Arcs represent relationships between elementary concepts or navigation requirements, e.g. to learn an abstract concept a sequence of elementary concepts must be learned. In simple cases each Presentation Description could represent an abstract concept. The navigation of an abstract concept is composed of the set of instantiated presentation units (pages) and the set of followed links. So a concept can be viewed in different ways, on the basis of user's browsing, profile and local context.
3. *Application Domain*. Finally, an Application Domain, i.e. an Adaptive Hypermedia, is composed by a set of Elementary Abstract Concepts organised in a digraph. Arcs represent relationships between EACs. An Application Domain is described as a set of Elementary Abstract Concepts and a set of links between them.

### 2.3. XML Presentation Descriptions and Metadata

In the proposed model, each data source is "wrapped" by an XML meta-description. As said before, the use of metadata is a key aspect for the support of multidimensional adaptation; furthermore, by means of meta-descripions, 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* for the XML meta-descriptions have been designed. They comprise descriptions of:

- *text*, hierarchically organized;
- object-relational *database tables*;
- *queries versus object-relational data*;
- *queries versus XML data*, expressed in *XML-QL* or *Xquery*;
- *video sequences*;
- *images*;
- *XML documents*;
- *HTML documents*.

As an example, consider the following meta-description of a query versus a relational database:

```
<query alias="trailersquery"
       IP-address="…"
       database-name="…"
       connection-driver="…" port="…"
       username="…" password="…"
>
  <SQL-statement>
    select video form trailers
    where key=#key
  </SQL-statement>
  <column name="video" type="ORDVideo"/>
</query>
```

In such meta-description, besides attributes regarding the connection to the DBMS, the key elements are the SQL statement (eventually with some parameters, `#key` in the example) and a description of the columns of the resulting table. Here, the only column contains objects of a complex type representing a MPEG video.

As said before, also Presentation Descriptions are XML documents. They are valid w.r.t. a DTD, whose key elements are the following:

```
<!ENTITY % dimension-parameters
  profiles NMTOKENS #IMPLIED
  network NMTOKENS #IMPLIED
  terminal NMTOKENS #IMPLIED
  QoS NMTOKENS #IMPLIED
  language NMTOKENS #IMPLIED
  location NMTOKENS #IMPLIED
  time NMTOKENS #IMPLIED
>

<!ELEMENT presentation-description
  (content|fragment|embedded-code|...)+>
<!ATTLIST presentation-description
  title CDATA #IMPLIED>

<!ELEMENT content (...)>
<!ATTLIST content
  link-destination CDATA #IMPLIED
  %dimension-parameters;
>

<!ELEMENT fragment EMPTY>
<!ATTLIST fragment
  reference CDATA #REQUIRED
  link-destination CDATA #IMPLIED
  %dimension-parameters;
>

<!ELEMENT embedded-code (#PCDATA)>
<!ATTLIST embedded-code
  link-destination CDATA #IMPLIED
  %dimension-parameters;
>
```

The XML PD is made up of 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 key elements of the PDs are *i)* the `content` element, which is used to include *text* in the pages, *ii)* the `fragment` element, useful for including basic multimedia fragments referenced by their aliases, and *iii)* the `embedded-code` element, which increases the flexibility of the pages allowing the insertion of terminal-dependent code (e.g. WML, HTML) in the page (obviously, wrapped by an XML `CDATA` section).

The `dimension-parameters` can be any XML `NMTOKEN`, and the author is allowed to decide his/her alphabet of labels regarding such parameters. This means that the model does not set up in advance the domains of the adaptivity dimensions, and it is always possible to extend them for a specific Application Domain. As an example, if the author provides different versions of a page on the basis of user's language, he/she can set the possible values of the variable regarding it; furthermore, the actual value of the variable could be forced by the user or inferred from his/her location.

### 2.4. Modelling Phases

The modelling of an adaptive hypermedia comprises the following phases:

- *Semi-automatic metadata creation*. Since basic multimedia information fragments are always accessed by means of metadata associated to them, the first phase is concerned with the creation of such metadata. The proposed architecture (see Sec.4) features a semi-automatic way to describe data sources, by means of an automatic exploration of some of them integrated by knowledge provided by the author.
- *High-level structure definition*. The author first defines the set of stereotype user profiles representing users' groups. Subsequently, he/she describes the overall Application Domain as a directed graph of EACs; here, some hints about typical graph structures are supplied. The link structure of the digraph representing the Application Domain is differentiated w.r.t. user profiles; thus, links between EACs naturally characterize feasible paths or navigation requirements among macro-concepts, distinguished on the basis of some characteristics of users' groups. In turn, the structure of single EACs represents paths between micro-concepts captured by single pages (Page Concepts); the author here differentiates links w.r.t.

user profiles and adds to them the probabilistic weights. It should be noted that the procedural (navigational) approach in the definition of the high-level structure of the Application Domain allows a simpler implementation with respect to declarative modelling approaches. The hypermedia designer can use a top-down approach, starting from the definition of the stereotype user profiles, or a bottom-up approach, first defining the page concepts and the link structures of EACs.

- *Simulation and validation of the probabilistic structure*. Since generally it is fundamental for an author to validate the high-level link structure of the hypermedia w.r.t. the mechanisms that drive the profile assignment decision, the proposed system provides a tool that allows the author of the hypermedia to examine in advance the "static" probabilistic properties of the hypermedia (see Sec.3) and to simulate the response of the system in the profile assignment decision.
- *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 dimension variables; thus, the author defines feasible sub-spaces of the overall adaptation space, to which correspond as many views over the Application Domain (see Sec.2.1). In our system this phase is performed essentially in a "visual" way: the author graphically composes the PDs; moreover, while accessing metadata the system allows to eventually enrich them, on the basis of the author's knowledge.

## 3. Probabilistic Hypermedia Scheme and User Model

In our model, an AD with $M$ different profiles is a set $N$ of XML documents where the generic document $i \in N$ contains, for each profile $k=1, \ldots, M$, a set $L_{ik}$ of annotated outgoing links $(i, j, k)$ where $j$ is the destination node. It can be mapped in a multigraph $G$ where each node corresponds to a XML document and each directed arc to an outgoing link:

$$G = ( N , E ), \qquad E = \bigcup_{\substack{i \in N \\ k=1,\ldots,M}} L_{ik}$$

For the sake of simplicity, the multigraph $G$ can be referred to as the set of the directed 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*.

$$G_k = ( N_k , E_k ) ,$$
$$N_k = \{ \, i \mid (i, j, k) \in E \vee (j, i, k) \in E \, \},$$
$$E_k = \{ \, (i, j) \mid (i, j, k) \in E \, \}$$

The proposed probabilistic approach assumes that the weight $W_k(i,j)$ of the arc $(i, j)$ in $E_k$ is the conditional probability $P(j|k,i)$, namely the probability that a user belonging to the profile $k$ follows the link to the $j$ node having already reached the $i$ node:

$$W_k(i, j) : E_k \rightarrow [\, 0, 1 \,]$$

$$W_k(i, j) = P(j|k,i), \quad (i, j) \in E_k, \quad k=1, \ldots, M$$

$P(i|k,i)$ 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.

$$\forall i \in N_k \quad \sum_{j \in N_k} W_k(i, j) = 1, \qquad k=1, \ldots, M.$$

A path $S$ in $G_k$ is defined as an ordered set of nodes:

$$S = \{ \, S_0 , S_1 , \ldots , S_l \mid (S_j, S_{j+1}) \in E_k, \; j=0, \ldots, l\text{-}1 \, \}.$$

We do not use the standard arc-based definition of a path because relaxing the condition $(s_j, s_{j+1}) \in E_k$ allows to consider a path involving different Logical Navigation Graphs. This could happen if a user in the profile $k$ chooses a link from a node $s_j$, to a node $s_{j+1}$ and he/she is moved to a new profile $h$; in that case we refer to $G$ and consider the alternative condition $(s_{j+1}, s_{j+2}, h) \in E$.

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. 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 $k$. The "shortest" path for each profile can be computed once.

Some *intrinsic properties* of the hypermedia structure can be expressed, for each profile $k$, by the following values:

- The medium of the probability of the minimum paths in $G_k$; high values of this term indicate the existence of highly natural paths in the hypermedia.

- The medium of the length of the minimum 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;
- The number of nodes belonging to profile $k$.

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.

In the proposed system, these properties are taken in account constructing three probability density functions (PDFs) whose values are proportional to them:

$$\mu(k)=\frac{\sum_{q=1}^{M}\left[\frac{\sum_{(i,j)\in E'_q}\tilde{P}_{ij}^{q}}{\left|E'_q\right|}\delta(k-q)\right]}{\sum_{q=1}^{M}\frac{\sum_{(i,j)\in E'_q}\tilde{P}_{ij}^{q}}{\left|E'_q\right|}}$$

$$n(k)=\frac{\sum_{q=1}^{M}\left[\left|N_q\right|\delta(k-q)\right]}{\sum_{q=1}^{M}\left|N_q\right|}$$

$$p(k)=\frac{\sum_{q=1}^{M}\left[\frac{\sum_{(i,j)\in E'_q}\left|\tilde{S}_{ij}^{q}\right|}{\left|E'_q\right|}\delta(k-q)\right]}{\sum_{q=1}^{M}\frac{\sum_{(i,j)\in E'_q}\left|\tilde{S}_{ij}^{q}\right|}{\left|E'_q\right|}}$$

where $E'_k$ is the set of arcs in the transitive closure of $G_k$. Then 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 $\beta_i>0$.

The probabilistic User Model collects information about the user's actions to build a discrete probability density function $A(k)$, with $k$=1, …, $M$, measuring the "belonging probability" of the user to each profile (i.e. how much each profile fits him/her). During the user's browsing activity the system updates $A(k)$ and the user's profile is changed consequently. So, on the basis of the user's behaviour, the system dynamically attempts to assign the user to the "best" profile. Moreover, the user can also "force" the changing of his/her profile.

Browsing starts from the presentation unit associated to a starting node. If the user is 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; 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 $s(k)$, then it decides the (new) profile to be assigned to the user. The page corresponding to the $R_r$ node is then generated in the computed profile. 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 main variables are:

- 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 evaluates, for each profile $k$:

- $P_R^k$, the probability of having followed the $R$ path through arcs belonging to the profile $k$;
- $\tilde{P}_{R_1,R_r}^k$, the reachability of the next node $R_r$ starting from the first node $R_1$, through arcs belonging to the profile $k$;
- $D^t[k]$, the distribution with respect to the profile $k$ of the visited nodes from $R_1$ to $R_{r-1}$, weighted with the time spent on each of them. 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 $D^t[k] = [\ (k_1, t_1+t_3), (k_2, t_1+t_2), (k_3, t_2), (k_4, t_3)\ ]$. The visiting times should be accurate; an interesting approach for an accurate computation is proposed in [21].

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 $k$. The reachability $\tilde{P}_{R_1,R_r}^k$ of the next node starting from the first node in $R$ takes in account the way the user *could have reached* that node. In fact, a high reachability of $R_r$ in the profile $k$ means the user would have reached the next node in a more "natural" way by following the links of the profile $k$.

Temporary deviations that do not move the user's interests can be taken in account trading off the effects of $P_R^k$ and $\tilde{P}_{R_1,R_r}^k$ on $A(k)$. The former takes in account the actual path so 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 $R_r$.

The distribution $D^t[k]$ shows how the time spent on visited nodes is distributed with respect to each profile; it is obviously an indicator of the interest the user has shown, with respect to the various profiles.

To avoid an "infinite memory" effect, only the most recently followed $r$-1 links ($r$ nodes) are considered. As an example, if $R$ was the path followed since the initial node, the probability $P_R^k$ of having followed $R$ in the profile $k$ would be zero if the user has visited just one node not belonging to the profile $k$. Note that we consider $W_k(i, j) = 0$ if $(i,j) \notin E_k$, $k$=1, …, $M$.

The evaluated variables are taken in account constructing three PDFs whose values are proportional to them:

$$c(k) = \frac{\sum_{i=1}^{M}\left[P_R^i \delta(k-i)\right]}{\sum_{i=1}^{M} P_R^i}, r(k) = \frac{\sum_{i=1}^{M}\left[\tilde{P}_{R_1,R_r}^i \delta(k-i)\right]}{\sum_{i=1}^{M} \tilde{P}_{R_1,R_r}^i}, t(k) = \frac{\sum_{i=1}^{M}\left[D^t[i]\delta(k-i)\right]}{\sum_{i=1}^{M} D^t[i]}$$

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 $\alpha_i > 0$.

### 3.1. The Assignment of Users to Profiles

The main idea to evaluate the belonging of a user to a profile is to combine his/her dynamic behaviour, synthesised in the following term *d(k)*, with the structural properties of the hypermedia scheme *s(k)*, mainly depending on its topology. The algorithm to compute the new PDF $A'(k)$ on the basis of the user's actions has the following structure:

Input:

- The discrete PDFs $A(k)$, $A_0(k)$ and $s(k)$;
- The recently followed path $R = \{R_1, \ldots, R_{r-1}, R_r\}$, composed by the last $r$ nodes visited by the user;
- The time spent on recently visited nodes, $t(R_1)$, ..., $t(R_{r-1})$.

Output:

A new probability density function $A'(k)$.

Steps:

1. Compute the new discrete PDF $d(k)$;
2. Compute the new discrete PDF $A'(k)$ as follows:

$$A'(k) = \frac{\gamma_0 A_0(k) + \gamma_1 A(k) + \gamma_2 d(k) + \Delta\gamma_3 s(k)}{\gamma_0 + \gamma_1 + \gamma_2 + \Delta\gamma_3}$$

where $\Delta$ = 1 if $s(k)$ has changed, $\Delta$ = 0 else.

We compute the new $A'(k)$ as a weighted medium of four terms; in particular, the first term expresses the initial user choices, the second term considers the story of the interaction, the third term captures the dynamic of the single user whereas the last term expresses "structural" properties of the hypermedia. 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

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

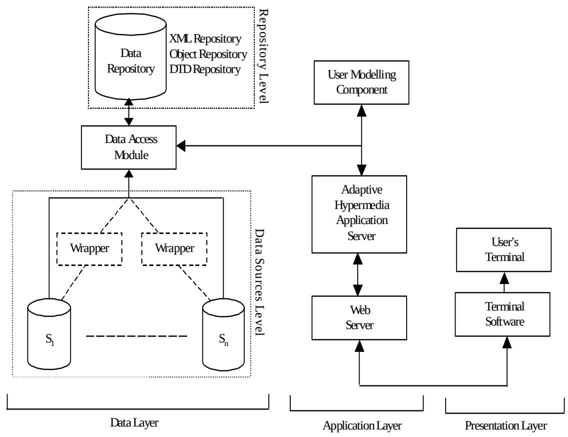


**Fig. 2.** System architecture

The Presentation Layer receives final pages to be presented and eventually scripts or applets to be executed; these scripts are useful for detecting e.g. local time, location, available bandwidth or evaluating 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).

At the Application Layer there are two main modules: the *Adaptive Hypermedia Application Server* (*AHAS*) and the *User Modelling Component* (*UMC*); 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.

The AHAS makes use of XSP technology. In fact, from XML descriptions of presentations, XSP pages are generated and stored in the XML repository. These generations are performed by means of XSL *Logicsheets*: the original XML file's elements are replaced with pieces of logic throughout an XSL transformation, the results of which is a canonical XSP page containing *logic* and *expression* elements etc.

Before delivering the final page to the user, the XSP page is executed (actually, it is previously compiled in a programming language) and a terminal-dependent XSL *Stylesheet* is applied to it.

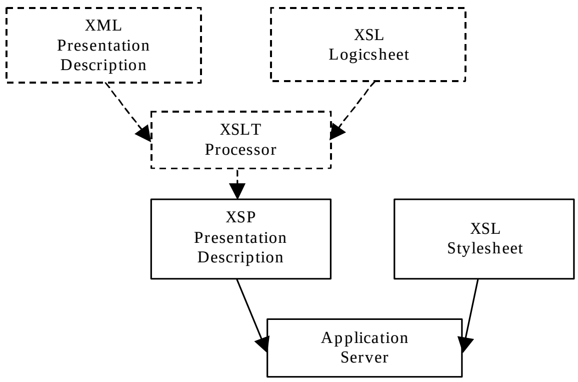


**Fig. 3.** Use of XSP at the Application Layer

Finally, the Data Layer stores persistent data and offers 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 is actually part of the Author Module, see Sec.4.2), 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 an *XML Repository*; these documents include XML Presentation Descriptions, generated XSP Presentation Descriptions, XSL logicsheets and stylesheets and XML metadata.
- persistent objects into an *Object Repository*; the objects represent logical navigation graphs and data about registered users.
- the DTDs used to validate XML documents.

Finally, the Data Access Module implements an abstract interface for accessing the data sources and the repository levels.

### 4.2. The Author Module

The Author Module (Fig. 4) has been designed to efficiently support the author of the hypermedia in both the structure definition and content composition phases, described in Sec.2.4. The main components of the Author Module are:

- The *Hypermedia Modeller*, which allows to design, in a visual way, the adaptive hypermedia as a digraph of EACs, and each EAC as a weighted digraph of Presentation Descriptions. In particular, it allows defining the probabilities of the arcs 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 w.r.t. syntax and semantics 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 in the Object Repository. As said before, the use of persistent representation allows reusing parts of the hypermedia; thus, after having been validated and stored, objects can be imported by the Hypermedia Modeller.
- The *Fragments Browser/Composer*, which allows browsing the information fragments provided by the Data Sources Level and accessing the XML metadata, using the Wrapper software components.
- The *PD Editor*, which allows the editing of XML Presentation Descriptions in the form of pure text, graphically as 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.
- The *PD Validator*, which performs a validating parsing of XML presentation descriptions with respect to the DTDs and stores them into the repository.
- An XSLT Processor, which applies the XSL logicsheets to the XML documents describing Presentation Descriptions and yields XSP documents (see also Fig.3).

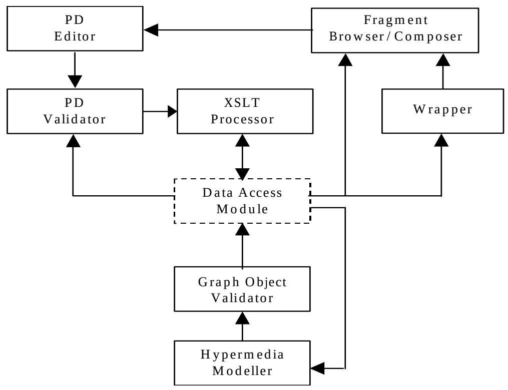


**Fig. 4.** The Author Module

#### 4.2.1. AHS Simulator and Validator

The Simulation Tool (Fig. 5) allows the author of the hypermedia to examine in advance the response of the UMC, on the basis of different kinds of users supposed to interact with the system (i.e. to move within its probabilistic structure).

By means of the Simulation Tool the author can:

1. analyse the intrinsic properties of the hypermedia calculated from its structure;
2. define a set of *User Classes* that describe the behaviour of typical users; many different *User Masks* can be assigned to each class, so the behaviour of each user can change during the same interaction with the system;
3. analyse the response of the UMC w.r.t. the User Classes.

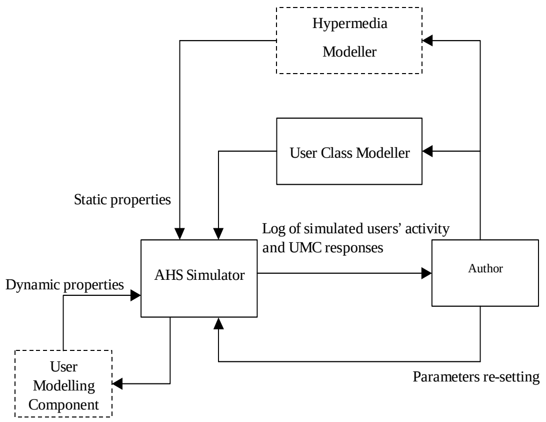


**Fig. 5.** The Simulation Tool

The main module of the Simulation Tool is the *AHS Simulator*. It is a multithreaded machine that serves the requests of the simulated users (threads generated by a *User Generator*), communicating them to the User Modelling Component, and storing the history of such requests, together with the response of the UMC.

At the end of the simulation (i.e. when the temporal duration is completed), the AHS Simulator shows, also graphically, the logs of simulated users' activity and the history of the discrete probability density functions measuring the "belonging probabilities" (or their mean values in the case of non-deterministic users); it allows to analyse their trend w.r.t. profiles (discovering e.g. if in some points of the hypermedia users are likely to be moved too frequently within profiles) or time (e.g. how the belonging probability to a profile has evolved over time).

Furthermore, after having run a simulation and examined the results, the author can decide to change the overall structure of the hypermedia, the length of the sliding temporal window, or the values of the parameters used to weight the probability density functions.

## 5. Conclusions

In this paper we presented an XML-based modular architecture for the modelling and the run-time support of web-based Adaptive Hypermedia Systems. The Adaptive Hypermedia is described as a three-dimensional space along the *User's behaviour, Technology,* and *External environment* adaptivity dimensions. So, the adaptation process is implemented finding the proper position of the user in that space, loading and adapting the corresponding XML page, applying to it the constraints bound to that point.

Currently we are completing the system implementation and its extensive experimentation through both the simulator and real tests, to obtain a complete environment for adaptive hypermedia design, simulation and deployment. The data centric approach in the basic data fragments representation and the use of orthogonal concepts in the modelling phase is currently used to implement terminal and network-independent hypermedia systems. In future works we intend to use such concepts as a basis to implement *environment-aware* web-based systems, such as the *location-dependent* services of mobile systems.

## 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. Apache Software Foundation, "The Apache Server project", http://httpd.apache.org.
3. Apache Software Foundation, "The Apache XML Project – Cocoon", http://xml.apache.org/cocoon.
4. Apache Software Foundation, " The Apache XML Project – Xalan 2", http://xml.apache.org/xalan.
5. Apache Software Foundation, "Cocoon's XSP processor", http://xml.apache.org/cocoon/xsp.html.
6. Apache Software Foundation, "The Jakarta Project - Tomcat", http://jakarta.apache.org/tomcat/.

7. Ardissono, L., et al., “A configurable system for the construction of adaptive virtual stores”, *World Wide Web Journal*, Baltzer Science Publisher, 1999.
8. Brusilovsky, P., “Methods and techniques of adaptive hypermedia”, in *User Modeling and User Adapted Interaction*, v.6, n.2-3, 1996.
9. Cannataro, M., Pugliese, A., “An XML-based architecture for adaptive web hypermedia systems using a probabilistic user model”, in *Proceedings of the IEEE IDEAS Conference*. IEEE Computer Society Press, 2000.
10. Cannataro, M., Cuzzocrea, A., Pugliese, A., “A probabilistic approach to model adaptive hypermedia systems”, in *Proceedings of the International Workshop for Web Dynamics*, 2001.
11. Cannataro, M., Cuzzocrea, A., Pugliese, A., “A probabilistic adaptive hypermedia system”, in *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2001.
12. Ceri, S., Fraternali, P., Bongio, A., “Web Modelling Language (WebML): a modelling language for designing web sites”, WWW9 Conference, 2000.
13. Ceri, S., Fraternali, P., Paraboschi, S., “Data-driven one-to-one web-site generation for data-intensive applications”, in *Proceedings of the 25$^{th}$ VLDB Conference*, 1999.
14. 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.
15. P. De Bra, P., Brusilovsky, P., Houben, G.J., “Adaptive Hypermedia: from systems to framework”, *ACM Computing Surveys, Symposium Issue on Hypertext and Hypermedia*, 1999.
16. De Bra, P., “Design issues in adaptive web-site development”, in *Proceedings of the second workshop on adaptive systems and user modeling on the WWW*, 1999.
17. Diestel, R., *Graph Theory*, Springer-Verlag, New York, 2000.
18. Garzotto, F., Paolini, D., Schwabe, D., “HDM – A model-based approach to hypermedia application design”, *ACM Transactions on Information Systems*, Jan., 1993.
19. Garzotto, F., Mainetti, L., Paolini, P., “Hypermedia application design: a structured approach”, in Schuler, W., Hannemann, J., Streitz, N., *Designing user interfaces for hypermedia*, Springer-Verlag, 1995.
20. Maes, P., “Software agents: humanizing the global computer”, in *IEEE Internet Computing*, v.1, n.4, 1997.
21. Murtagh, F., Tao, F., “Towards knowledge discovery from WWW log data”, in *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2000.
22. Schwabe, D., Rossi, G., “The Object-Oriented Hypermedia Design Model”. *CACM* 38(8): 45-46, 1995.
23. Schwabe, D., De Almeida Pontes, R., “OOHDM-WEB: Rapid prototyping of hypermedia applications in the WWW”, Technical Report, Pontifícia Universidade Católica do Rio de Janeiro, 1998.