A Probabilistic Approach to Model Adaptive Hypermedia Systems

Mario Cannataro, Alfredo Cuzzocrea, Andrea Pugliese

ISI-CNR, Via P. Bucci, 41/c
87036 Rende, Italy
cannataro@si.deis.unical.it, {alfredo.cuzzocrea,andreapugliese}@tin.it

Abstract. Web-based hypermedia systems allow user-driven access to information and services and content personalization. Application fields where adaptive hypermedia systems could be used are on-line learning, electronic commerce, on-line advertising etc. This paper presents a probabilistic approach for the modelling of Adaptive Hypermedia Systems (AHS). The Application Domain is modelled along three different adaptivity dimensions: technology (network and user's terminal), user's behaviour and external environment (location, language, socio-political issues, etc.). The user's behaviour is modelled using a probabilistic approach and the most promising profile, that is a "view" over the application domain, is dynamically assigned to the user, using a discrete probability density function. A modular architecture supporting both the authoring phase and the run-time support of the Adaptive Hypermedia System is also presented. The authoring module allows the description of the Application Domain, the dynamic data access and data composition functions, extracting pre-existing multimedia basic data from different data sources. The basic data fragments are described through XML-based metadata.

1. Introduction

Web-based hypermedia systems are becoming increasingly popular as tools for user-driven access to information and services. The linking mechanism of hypermedia offers users a large amount of navigational freedom so that it becomes necessary to offer support during navigation. Moreover the classes of users to be dealt with are becoming increasingly heterogeneous and demanding due to the worldwide deployment of applications. Typical application fields for hypermedia systems are on-line learning and teaching, electronic commerce and on-line advertising.

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's behaviour and to the content provider's goals. The Application Domain Model is used to describe the hypermedia basic contents and their organisation to depict more abstract concepts. The User Model attempts to describe the user's characteristics and preferences and his/her expectations in the browsing of hypermedia. The adaptation of the Application Domain content presentations to the User Model can be variously dynamic, and different schemes are described in the next Section.

To efficiently allow the realisation of user-adaptable contents and presentation, a clear separation between basic multimedia contents, domain model and User Model should be achieved. Whereas the conceptual model offered by standard database technologies (relational or object-oriented) could 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 allow an efficient access to basic data fragments, their dynamic selection, extraction and composition. Methodologies to track, in a non-invasive way, the user's behaviour and to mine the log data should be applicable.

This paper presents the design of an Adaptive Hypermedia System (AHS) that uses a probabilistic approach for the modelling of the Application Domain data and the user's behaviour, extending the work in [CP00].

The proposed Application Domain Model uses a layered data model, distinguishing among Elementary Abstract Concepts, Presentation Descriptions (nodes) and basic data fragments. PDs are organized in a weighted multigraph into which, for each supported user's *stereotype profile* (which indicates the user's belonging to a *group*), the arc's weight represents the probability to follow the corresponding link, i.e. the probability to reach the next Presentation Description. At the lowest level each Presentation Description is an XML document [W3C00, Walsh98]: XML data-centric orientation makes it 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).

The Application Domain is modelled along three different adaptivity dimensions: *technology* (e.g. network conditions and user's terminal), *user's behaviour* (browsing activity) and *external environment* (location, language, socio-political issues, etc.). These dimensions are monitored collecting a set of values, called *User*, *Technological* and

External Variables; in this way it is possible to model the adaptivity dimensions, and a probabilistic User Model is constructed.

In particular, using a sort of sliding temporal window, the system takes in account the recent path followed by the user along the graph and the alternative shortest paths (i.e. the paths with highest probabilities for each profile), evaluating joint probabilities for each profile. Moreover, the visited nodes are clustered with respect to their most relevant profile.

These values are used in conjunction 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 chooses to leave the user within the current profile or to move him/her to a new profile. So dynamically, on the basis of the user's behaviour, the system attempts to assign the user to the "best" profile (i.e. a particular "view" over the Application Domain) that fits his/her expectations. The user's profile and External variables drive the generation of a presentation unit, that is formatted and presented according to the technological variables.

The rest of the paper is organised as follows. Section 2 summarises Adaptive Hypermedia Systems. Section 3 describes the graph-based Application Domain Model. Section 4 presents a probabilistic approach to model the user's behaviour and a method to dynamically assign a user to an available profile. Section 5 describes a multi-tier architecture supporting such a model. Section 6 contains conclusions and outlines future work.

2. Adaptive Hypermedia Systems

In the last years some frameworks to build Adaptive Hypermedia Systems have been developed. As said before, basic components of an AHS are the Application Domain Model, the User Model and the methods to adapt presentations to the user. The most promising approach in modelling the Application domain is data-centric, and researches are made to adapt well known modelling techniques used in databases to hypermedia systems [CFP99, Abi&al99].

The adaptation of the Application Domain presentation to the User Model follows various schemes that can be summarised as follows:

- In *adaptable* hypermedia, the user provides a profile, and the contents are delivered with respect to the selected profile;
- In *adaptive* hypermedia, the system monitors the user's behaviour and adapts the presentation accordingly. The user's behaviour is often deducted from his/her browsing activity;
- In *dynamic* hypermedia there are no predefined presentations, so on the basis of the user's behaviour (i.e. some attribute values) the system generates presentations combining in a dynamic way atomic components, such as images, texts, sounds and so on.

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 [Brus96]. In the construction of the profiles, many different aspects are generally taken in account: user's *knowledge* of the Application Domain to deal with, *goals* and *preferences* that guide his/her browsing, or the *reading key* the user wants to obtain from the hypermedia.

Profile updating (in the case of overlay models) or an assignment of the user to a profile (in the case of stereotype models) must be made dealing with some information the user "shares" with the system [Kay95]. These information can be collected directly involving the user (*co-operative User Modelling*) or just observing his/her actions (*automatic User Modelling*). Automatic User Modelling is obviously not completely reliable, so it can be useful to let the user change his/her profile "on the fly", giving him/her an hint based on automatic evaluations if necessary.

Adaptation can be generally distinguished into *adaptive presentation* and *adaptive navigation support*. Adaptive presentation is a manipulation of fragments: they can be conditionally included, their order can be changed or they can be made "less visible" in the pages. Another interesting technique is based on the use of *stretchtexts*, e.g. text fragments that can be also presented as short placeholders; the initial presentation of fragments is decided by the system while the user can "stretch" or "shrink" fragments on the basis of his/her interests.

Adaptive navigation support is a manipulation of links: a "direct guidance" (e.g. a "next" button) can be featured, links can be conditionally included, sorted, disabled, or annotated on the basis of their importance; their structure can be presented providing an adapted map of the hypermedia.

Some recent adaptive hypermedia systems are outlined below:

- The *AHA* system [DC98] uses overlay profiles with boolean attributes, which are updated presenting tests to the user (for tracking *knows-about-something* concept) or logging the pages he/she has read (for tracking *read-about-something* concept). The system conditionally includes fragments using the boolean values, and supports link annotation by means of link classes that are presented into different colours.

- The *Torii* system [CFP99] dynamically assigns a stereotype profile to the user with *Event-Condition-Action* rules, on the basis of his/her behaviour, and *site views* corresponding to stereotypes are presented. Moreover, in the Structural Level the aggregation of presented data (and concepts in general) can be differentiated with respect to each individual user.
- The *AVANTI* project [FKN96] acquires *primary assumptions* about the user with an initial interview and observing certain dialog actions (e.g. a request of an explanation); based on them draws inferences on further assumptions.
- In [ACT98], in a news personalization system, on the basis of a questionnaire an initial stereotype profile is assigned to the user by means of a set of probabilistic rules; these rules are used also for the updating of a dynamic overlay profile.
- The *SETA* system [Ard&al99] constructs adaptive hypermedia useful to guide the user in the exploration of a product catalogue in a virtual store, on the basis of his/her overlay profile.
- In [HH98] three different systems are evaluated; one of them does not support any adaptation, while the others implement respectively the adaptive presentation and adaptive navigation support.
- *InterBook* [BE98], a non-Web based education-oriented system, supports link annotation changing icons and fonts of the text anchor on the basis of the user's knowledge.
- *MetaDoc* [BE94] uses stretchtext technique to present personalized views and to track the user's interests.
- *Syskill & Webert* [BP97] is an application which extends the browser and constructs a User Model on the basis of some scores he/she assigns to visited pages. On the basis of this model, a link annotation adaptation is performed, and a more accurate Web contents' search is achieved.
- The e-commerce-oriented *TELLIM* system [Joerding99] maintains data about *interest* the user shows about available products; these data are used to compute a set of "prediction" rules.

  Some interesting systems, mainly based on Data Models derived from databases are:
- The *Strudel* system [Fern&al98] applies a *site-definition query* to the underlying data to define the web-site structure, thus resulting in a "site graph", which represents both the site's content and structure. The author specifies the visual presentation of pages in Strudel's HTML-template language.
- The *Araneus Data Model* is explained in [AMM97]; the model is inspired to the typical web-site structures. On its basis, a language called *Ulixes* is used to build database views of the Web, and a language called *Penelope* is used to define hypertexts from relational views.
- *Active Views* [Abi&al99] is a system oriented to electronic commerce which offers a declarative view specification language to describe a web application that allows users to work interactively on the specified data in a standard distributed environment.

3. Application Domain modelling

This Section presents a probabilistic model to describe the Application Domain using directed multigraphs of XML documents. After a brief description of the parameters to be considered to effectively support the different adaptivity goals/requirements, a probabilistic model to describe adaptive hypermedia is presented.

In the probabilistic Application Domain Model, that uses a layered data model, the hypermedia's links are weighted with the probability that a user visiting a node will follow one of them. The model provides *M* predefined *stereotype profiles* to which correspond as many *Logical Navigation Graphs*.

3.1 Dimensions of adaptivity

The goal of AHS is to adapt contents and presentations to satisfy the user's goals. Some of these goals can be captured analyzing the user's behavior (one or many), 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 per-user/location available network bandwidth can allow to satisfy response-time requirements. Many of these user wishes do not depend on the user's behaviour, and can be considered as orthogonal requirements.

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

- *Technology* (network bandwidth, Quality of Service, user terminal, etc.);
- *User's behaviour* (per-user behaviour, collective behaviour, etc.);
- *External environment* (location, language, socio-political issues, external web sites content, etc.).

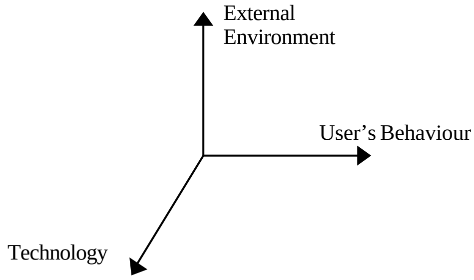

Fig. 1. Adaptivity dimensions

The AHS monitors the different possible sources that can affect those dimensions, collecting a set of values, called *User*, *Technological* and *External Variables*. The choice of what variables to consider depends mainly on the application domain, although some common aspects can be specified. The adaptation strategy could be implemented using a (non-deterministic) decision-tree or a rule-based approach.

The user's profile and External variables mainly drive the generation of page content, while technological variables mainly guide the adaptation of page layout. For example, an e-commerce web site could show a class of products that fits the user needs (deducted by his behaviour), applying a time-dependent price (night /day), formatting data w.r.t. the user terminal (e.g. WML for WAP mobile phones) and sizing data w.r.t. the network bandwidth.

3.2 A layered Application Domain Model

The proposed Application Domain Model uses a layered data model: at the top level an adaptive hypermedia is represented by a directed multigraph of Elementary Abstract Concepts (nodes); at the middle level, each Elementary Abstract Concept is described by a weighted directed multigraph of Presentation Descriptions (nodes). For each supported user's profile, the arc's weight represents the probability to follow the corresponding link, i.e. the probability to access the next Presentation Description. At the lowest level each Presentation Description is an XML document. XML data-centric orientation makes it 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).

The layered data model extends the Adaptive Data Model described in [DBH99]; it comprises the following abstract levels:

0. *Information Fragments* (or *atomic concepts*) like texts, sounds, images, videos, etc. at the lowest level. The information fragments are stored in databases, files or everywhere in the Web. 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). 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. For example, an image could be represented using different levels of details (sizes), formats, points of view (inquadrature), whereas a text could be organized as a hierarchy of fragments, represented using different languages.
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. The XML documents comprise the *fragments* and *content* Sections. The former contains a set of elements indicating fragments that will be extracted from the data sources to construct the page and defines some aliases for them. The latter describes how these fragments must be presented to the user, referring to the declarations made into the fragments Section. In this Section a set of internal links to other documents is also indicated, which are used to represent relationships or simply the next documents that should be visited. The links are annotated by a weight, that represents their importance with respect to each other, and that in our model is a probability (see Section 3).
   - 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 (e.g. using DOM APIs [W3C99]) 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 graph. 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 is composed by a set of Elementary Abstract Concepts organised in a graph. Arcs represent relationships between EACs. An Application Domain is described as a set of Elementary Abstract Concepts and a set of links between them; the data describing an Application Domain are stored in the repository.

The proposed model supports stereotype profiles, so an application domain is organised in such a way it is possible to regard it along different views corresponding to profiles. A user belonging to a given profile will see the corresponding view of the domain. A Presentation Description that is relevant to more profiles will be represented by a XML document (from here named node) containing all the corresponding fragments, contents and links, differentiated with respect to each profile.

In our system the modelling of an adaptive hypermedia comprises the following phases:

1. definition of H main stereotypes profiles, representing the main views through which to see the hypermedia; this phase is driven by the knowledge of the application domain;
2. definition of the overall Application Domain, as a directed multigraph of EACs, differentiated with respect to the main profiles;
3. for each main profile, definition of a (eventually empty) set of stereotype sub-profiles representing a specialisation of the hypermedia with respect to the behaviour of the current user; these sub-profiles are $M \geq H$.
4. definition of the structure of each EAC, differentiated with respect to the sub-profiles;
5. definition of the basic Presentation Descriptions, comprising multimedia contents, presentation layout and format, access rights to data (where applicable, e.g. when a presentation unit refers to a database table) etc. The presentation units can integrate heterogeneous data sources using *Wrapper* software components, which produce a neutral XML meta-description of information.

It should be noted that the procedural (navigational) approach in the definition of the Application Domain allows an easier implementation with respect to declarative modelling approaches. The hypermedia designer can use a top-down approach, starting from the definition of the main profiles and of the overall Application Domain, or a bottom-up approach, starting from the definition of the presentation description and of the atomic concepts they are composed of.

The adaptation of hypermedia contents, by means of transformation from *Presentation Descriptions* to *pages*, comprises both *adaptive presentation* and *adaptive navigation support*. The former is obtained instantiating each Presentation Description with respect to the user's profile, technological conditions and external variables (e.g. the same concept can be represented differently with respect to profiles either referencing different fragments or links). To support *adaptive navigation* the model, referring to the weighted multigraph-based representation of XML nodes, presents to user only relevant links. The rest of the Section describes the probabilistic approach used to model the hypermedia structure.

3.2.1 Probabilistic hypermedia scheme

An adaptive hypermedia is modelled by means of a directed multigraph of EACs. Each EAC is, in turn, represented by a weighted directed multigraph of Presentation Descriptions, i.e. XML documents. Although the structure of the hypermedia data at the levels 2 and 3 of our data model are essentially the same, the semantic interpretation of the inter-EAC navigation is different w.r.t. the intra-EAC navigation.

At this level the author has defined the M stereotype profiles through which the hypermedia can be viewed, whereas each EAC can be entered and viewed by using one of the M profiles (a specific EAC that is only relevant to one profile will have only a view, whereas a general EAC could have at maximum M views). Exiting from an EAC can be done having the same entering profile or a different one, mainly depending on the user's behaviour. The choice of the profiles is mainly driven by a knowledge of the application domain, and can be also dynamically driven by a classification of the users accessing the system, on the basis of a data-mining analysis of the log data.

In a similar way, an EAC is a set of XML documents, i.e. Presentation Descriptions, and each of them can be viewed using M different profiles. Moreover, these profiles allow to specialise the presentations not only on the basis of the author's guide, that as said before, is driven by general considerations, but mainly on the basis of the single user behaviour. So, it is also possible, for a given EAC, to have $M > H$ profiles.

In the rest of the Section we will define the graph structure of the EACs, whereas we omit the description of the overall adaptive hypermedia that can be directly obtained by applying the given definitions.

An EAC with M different profiles (reading keys), 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*.

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

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:

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

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.

It is appropriate to always show a not too poor set of links, to let the user easily reach the resources he/she is interested in. Moreover, it is important to avoid the *horizon narrowing* phenomenon [Maes97] so the graphs should not contain high probability cycles.

Similarly to the modelling of an EAC, an adaptive hypermedia AH, with H different stereotype profiles (reading keys), is a set of Q EACs where the generic node i ($i=1, \ldots, Q$) contains, for each profile $k=1,\ldots,H$, L_{ik} annotated outgoing links. It can be mapped in a multigraph where each node corresponds to an EAC and each directed arc to an outgoing link.

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 PDFs whose values are proportional to them:

$$m(k)=\frac{\sum_{i=1}^{M}\left[\frac{\sum_{(i,j)\in E'_k}\tilde{P}^k_{ij}}{\left|E'_k\right|}d(k-i)\right]}{\sum_{i=1}^{M}\frac{\sum_{(i,j)\in E'_k}\tilde{P}^k_{ij}}{\left|E'_k\right|}},\qquad n(k)=\frac{\sum_{i=1}^{M}\left[\left|N_i\right|d(k-i)\right]}{\sum_{i=1}^{M}\left|N_i\right|},\qquad p(k)=\frac{\sum_{i=1}^{M}\left[\frac{\sum_{(i,j)\in E'_k}\left|\tilde{S}^k_{ij}\right|}{\left|E'_k\right|}d(k-i)\right]}{\sum_{i=1}^{M}\frac{\sum_{(i,j)\in E'_k}\left|\tilde{S}^k_{ij}\right|}{\left|E'_k\right|}}$$

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:

$$s(k)=\frac{b_0 m(k)+b_1 n(k)+b_2 p(k)}{b_0+b_1+b_2}$$

where the values of $m(k)$ 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 $\beta_i>0$.

4. A probabilistic approach to adapt the hypermedia to the user's behaviour

The probabilistic User Model collects information about the user's actions to build a discrete probability density function (*PDF* in the following) $A(k)$, with $k=1, \ldots, M$, 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.

4.1 The probabilistic User Model

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, $t(R_1), \ldots, t(R_{r-1})$.

On this basis, the system evaluates, for each profile k:

- P^k_R, the probability of having followed the R path through arcs belonging to the profile k;
- $\tilde{P}^k_{R_1,R_r}$, 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 [MT00].

A high value of P^k_R 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 $\tilde{P}^k_{R_1,R_r}$ 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:

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

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 $\alpha_i > 0$.

4.2 Algorithm for the evaluation of the belonging probabilities

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 $s(k)$ (see Section 3);
- 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.

5. System architecture

A system supporting the probabilistic Application Domain and User Models has been designed and partially implemented. The system has a three-tier architecture comprising the *User, Application* and *Data Layer* (Fig. 2). *User layer* corresponds to the browser. It carries out "light" computation because it receives HTML pages and executes only an invisible applet that signals to the server that the user is not interested to that page at the moment (e.g. has reduced to icon its window).

Basic web technologies used to implement the system are either those concerning the on-line access to database, data composition and data delivery (e.g. JDBC [MagLa99b], Java Servlet [MagLa99], Enterprise Java Beans, XML), and those allowing client-side elaboration (e.g. Java Applet).

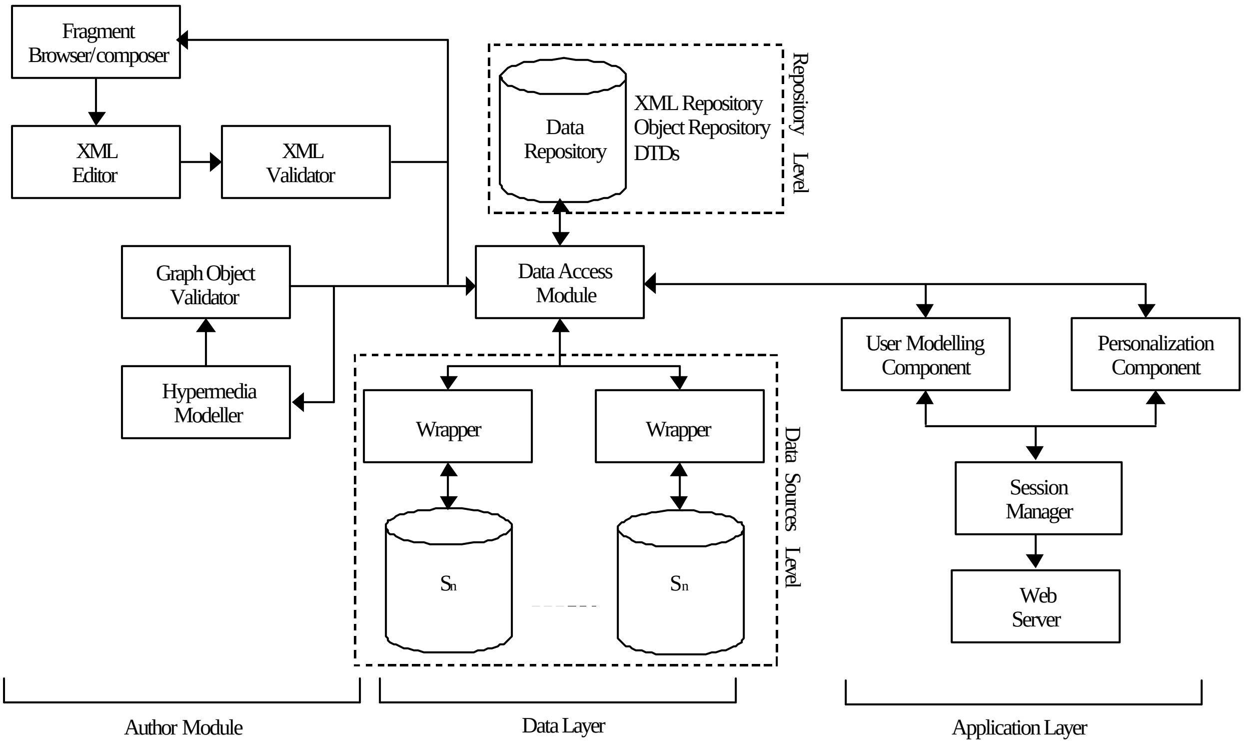

Fig. 2. System architecture (Application and Data Layers) and Author Module

5.1 Application layer

At the *Application Layer* (Fig.2) there are three main modules: *Session Manager* (SM), *User Modelling Component* (UMC) and *Personalization Component* (PC) [Ard&al99]; they run together with a Web-Server.

Session Manager maintains a separate session for each user. It responds to the user's actions and communicates them to the User Modelling Component. The UMC, which maintains the most recent actions of the user, executes the algorithm for the evaluation of belonging probabilities, evaluates the new profile, and gives the result back to the SM. With the new profile the SM asks to the Personalization Component for the presentation unit, instantiated with respect to that profile, to external variables and with respect to technological conditions. The PC dynamically generates the HTML code of the page; after having received the definitive page the SM sends it to the user.

In the final page the most likely links are generally more visible (e.g. showing a "next page" button or using different colours). Furthermore, it can be useful to maintain the most recent pages visited by the user in a cache, thus reducing the overhead of reconstructing recent HTML pages for users with the same profile, technological conditions and external variables.

The Personalization Component executes the following steps:

1. Extracts from the Data Repository the XML Presentation Description to be instantiated;
2. Browses the "content" Section and constructs the outgoing HTML page, referring to the aliases declared in the "fragments" Section and selecting them on the basis of User, Technological and External Variables.
3. Returns the definitive HTML page.

5.2 Data Layer

The main goal of the *Data Layer* (Fig.2) is to store persistent data and to offer efficient access primitives. It provides data to the Application Layer, allowing access to the logical navigation graphs, to XML documents and data fragments, respectively to the UMC and to the PC.

The Data Layer 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 implement the hypermedia. Each data source S_i is accessed by a Wrapper that also 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*. We choose to adopt the *XDM* (XML Data Model [FGZ00]) format to store the XML information as a set of relational records in a relational database (XMLDB);
- 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 useful for separating the modules which access data from the particular implementations of the Data Sources and Repository levels.

5.3 Author Module

The design of the adaptive hypermedia is carried out using an *Author Module*. It allows to design and validate (with respect to syntactic and semantic correctness) the XML documents representing the Presentation Descriptions and the persistent objects representing the Elementary Abstract Concepts and the overall hypermedia.

The main components of the Author Module are (Fig. 2):

- The *Hypermedia Modeller*, which allows to design the adaptive hypermedia as a multigraph of EACs. Moreover, it allows to design the multigraphs representing an EAC. In particular, it allows to define the probabilities of the arcs and offers a set of utilities such as: search of shortest (maximum weigth) path, minimum spanning tree, etc.
- The *Graph Object Validator*, which receives graph descriptions from the Hypermedia Modeller and, after a validation of them (e.g. coherence of probabilities, congruence with the links contained in the Presentation Descriptions etc.), generates persistent objects (e.g. Java Entity Beans) containing the weighted graphs and stores them in the Data Layer (Object Repository). The persistent objects contain all the data useful for the system computed at the generation time, e.g. the shortest path for each pair of nodes and the PDF $s(k)$ (see par.3.2.1); the utility emphasises the existence of some high-probability cycles and asks for confirm. The use of a persistent representation allows to reuse (part of) of the hypermedia, e.g. some EACs, for different web systems.
- The *Fragments Browser/Composer*, which allows the browsing of information fragments and XML Metadata, provided by the heterogeneous data sources or stored in the repository, and their aggregation to form more complex data (which are also represented by XML documents).
- The *XML Editor*, which allows the editing of XML documents (Presentation Descriptions and fragments metadata) in the forms of pure text, graphically as trees, or in a "visual" way. It is possible to create new documents, to edit pre-existing ones and to browse the available information fragments and metadata, by means of the Fragments Browser.
- The *XML Validator*, which performs a validating parsing of XML documents with respect to the DTDs and stores them into the repository.

6. Conclusions and future work

In this work we presented a probabilistic approach for the development of Adaptive Hypermedia Systems based on stereotype profiles. The architecture uses a layered multigraph structure to describe the application domain and a probabilistic model to adapt the web-site contents and links to the user's behaviour.

The user's behaviour is described using a probabilistic model: the main contribution of the User Model is to combine information regarding the followed path, the "shortest" path (i.e. the path not necessarily followed but with the highest probability) and a classification of the visited nodes, to calculate the distribution of the belonging probabilities. Moreover, the system takes into account the structural properties of the hypermedia also represented through a PDF. So, the most promising profile, that is a "view" over the application domain, is dynamically assigned to the user, using that probability distribution. The views over the domain model are currently static (with a fixed set of profiles) but it is possible to dynamically change the weights of the graphs' arcs on the basis of context change and user's behaviour.

We are developing a prototype of the system using Java [HC99] to enhance portability. Current and future work will concern the validation and tuning of the probabilistic model, the development of tools to improve and simplify the authoring phase, the full support of the personalization of the hypermedia structure on the basis of technology and external variables.

References

[Abi&al99] S. Abiteboul, B. Amann, S. Cluet, A. Eyal, L. Mignet, T. Milo, *Active views for electronic commerce*, in *Proceedings of the 25th VLDB Conference*, 1999.

[ACT98] L. Ardissono, L. Console, I. Torre, *Exploiting User Models for personalizing news presentations*, in *Proceedings of the 2nd workshop on adaptive systems and User Modeling on the WWW*, 1998.

[Ard&al99] L. Ardissono, A. Goy, R. Meo, G. Petrone, L. Console, L. Lesmo, C. Simone, P. Torasso, *A configurable system for the construction of adaptive virtual stores*, World Wide Web Journal, Baltzer Science Publisher, 1999.

[AMM97] P. Atzeni, G. Mecca, P. Merialdo, *To weave the Web*, in *Proceedings of the 23rd VLDB*, 1997.

[BE94] C. Boyle, A.O. Encarnacion, *Metadoc: An Adaptive Hypertext Reading System*, in *User Modeling and User-Adapted Interaction* 4(1): 1-19, 1994.

[BE98] P. Brusilovsky, J. Eklund, *The value of adaptivity in hypermedia learning environments: a short review fo empirical evidence*, in *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia*, 1998.

[BP97] D. Billsus, M. Pazzani, *Learning probabilistic User Models*, in *Proceedings of the Sixth International Conference on User Modeling*, 1997.

[Brus96] P.Brusilovsky, *Methods and techniques of adaptive hypermedia*, in *User Modeling and User Adapted Interaction*, v.6, n.2-3, 1996.

[Brus97] P. Brusilovsky, *Efficient techniques for adaptive hypermedia*, in: C. Nicholas and J. Mayfield (eds.): *Intelligent hypertext: Advanced techniques for the World Wide Web*. Lecture Notes in Computer Science, Vol. 1326, Berlin: Springer-Verlag, pp. 12-30, 1997.

[CFP99] S. Ceri, P. Fraternali, S. Paraboschi, *Data-driven one-to-one web-site generation for data-intensive applications*, in *Proceedings of the 25th VLDB Conference*, 1999.

[CP00] M. Cannataro, A. Pugliese, *An XML-based architecture for adaptive web hypermedia systems using a probabilistic User Model*, IEEE IDEAS 2000 Conference, 18-20 Sept. 2000, Yokohama, Japan. IEEE Computer Society Press, 2000.

[DeBra99] P. De Bra, *Design issues in adaptive web-site development*, in *Proceedings of the second workshop on adaptive systems and User Modeling on the WWW*, 1999.

[DC98] P. De Bra, L. Calvi, AHA: *A generic adaptive hypermedia system*, in *Proceedings of the 2nd workshop on adaptive hypertext and hypermedia*, Pittsburgh, 1998.

[DBH99] P. De Bra, P. Brusilovsky, G.J. Houben, *Adaptive Hypermedia: from systems to framework*, ACM Computing Surveys, Symposium Issue on Hypertext and Hypermedia, 1999

[Fern&al] M. F. Fernandez, D. Florescu, A. Y. Levy, D. Sucin, *Catching the boat with Strudel: experiences with a web-site management system*, in *Proceedings of SIGMOD'98*, 1998.

[FGZ00] S. Flesca, S. Greco, E. Zumpano, *Modelling and querying XML data*, IEEE IDEAS 2000 Conference, 18-20 Sept. 2000, Yokohama, Japan. IEEE Computer Society Press, 2000.

[FKN96] J. Fink, A. Kobsa, A. Nill, *User-oriented adaptivity and adaptability in the AVANTI project*, in *Designing for the Web: empirical studies*, Microsoft usability group, Redmond, 1996.

[HC99] C.S. Horstmann, G. Cornell, *Core Java 1.2*, vol. 1 e 2, Sun Microsystems Press, 1999.

[HH98] J. Hothi, W. Hall, *An evaluation fo adapted hypermedia techniques using static User Modelling*, in *Proceedings of the 2nd workshop on adaptive hypertext and hypermedia*, Pittsburgh, 1998.

[HL99] G.J. Houben, P.Lemmens, *A Software architecture for generating hypermedia applications for ad-hoc database output*, Technical Report, Eindhoven University of Technology, Department of Computing Science, http://wwwis.win.tue.nl/~houben/pub/csr99-16.ps.

[Joerding99] T. Joerding, *A temporary User Modeling approach for adaptive shopping on the web*, in *Proceedings of the Second Workshop on Adaptive Systems and User Modeling on the WWW*, 1999.

[Kay95] J.Kay, *Vive la difference! Individualized interactions with users*, in *Proceedings of the 14th IJCAI*, Montreal, 1995.

[Maes97] P. Maes, *Software agents: humanizing the global computer*, in *IEEE Internet Computing*, v.1, n.4, Luglio/Agosto 1997.

[MagLa99] MageLang Institute, *Fundamentals of JavaServlets*, Short course, 1999.

[MagLa99b] MageLang Institute, *JDBC-Java database programming*, Short course, 1999.

[MT00] F. Murtagh, F. Tao, *Towards knowledge discovery from WWW log data*, in *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2000.

[Nijholt99] A. Nijholt, J. Hulstijn, *Multimodal Interactions with Agents in Virtual Worlds*, in N. Kasabov (ed.), *Future Directions for Intelligent Systems and Information Science*, Phisica-Verlag: Studies in Fuziness and Soft Computing, 1999.

[Walsh98] N.Walsh, *What do XML documents look like?*, XML.com Tutorial, 1998.

[W3C00] World Wide Web Consortium, *Extensible Markup Language*, Recommendation, 2000.

[WHD99] H.Wu, G.J. Houben, P. De Bra, *Authoring support for adaptive hypermedia applications*, Ed-Media 99 Conference.