A Probabilistic Adaptive Hypermedia System

Mario Cannataro, Alfredo Cuzzocrea and Andrea Pugliese
ISI-CNR, Via P. Bucci, 41/c - c/o DEIS-University of Calabria, 87036 Rende, Italy
cannataro@si.deis.unical.it, [alfredo.cuzzocrea, andreapugliese]@tin.it

Abstract

Adaptive Hypermedia Systems allow user-driven access to information and services and content personalization. This paper presents the design and development of a Probabilistic Adaptive Hypermedia System that uses a probabilistic approach for user modelling and presentation adaptation. The Application Domain is modelled along three different adaptivity dimensions: user's behaviour (browsing activity), technology (network and user's terminal constraints), and external environment (time, 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 implementing both the authoring phase and the run-time support of the Adaptive Hypermedia System is also presented.

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. Moreover the design of a web based hypermedia should take into account the different classes of users that are becoming increasingly heterogeneous due to world-wide deployment, different interests, kind of terminals and social conditions.

To face these problems, in the last years the concepts of user modelling and adaptive graphical user interface have come together in the Adaptive Hypermedia Systems (AHS) research theme. Typical application fields for AHS are on-line learning and teaching, electronic commerce and on-line advertising. Some recent AHS are outlined in [7].

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 most promising approach in modelling the Application domain is data-centric, and many researches use well known database modelling techniques [1, 8].

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 [5].

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.

This paper presents the design and development of a Probabilistic Adaptive Hypermedia System (PAHS) that uses a probabilistic approach for the modelling of the Application Domain data and the user's behaviour, extending the work in [7, 9].

The proposed Application Domain Model uses a layered data model, where upper (abstract) layers are organised as weighted digraphs (multigraphs) of concept descriptions and lower (physical) layers are organised as XML [19, 20] documents referring to basic multimedia fragments stored in different data sources. In upper layers the arc's weight represents the probability to follow the corresponding link, i.e. the probability to reach the next concept.

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* and a probabilistic stereotype User Model is constructed, which expresses the user's belonging to a group [5]. The collected *user's behaviour* and *external environment* 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 (stereotype) profile is updated as long as the browsing goes on. Using that distribution 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. Moreover, the delivered presentation units are formatted and presented according to network and terminal constraints, and external environment conditions.

0-7695-1062-0/01 $10.00 © 2001 IEEE

The rest of the paper is organised as follows. Section 2 describes the Application Domain Model and the User Model of the Probabilistic AHS. Section 3 describes the system architecture. Finally, Section 4 contains conclusions and outlines future work.

2. A Probabilistic adaptive hypermedia system

The main components of the proposed Probabilistic Adaptive Hypermedia System are described in the following.

2.1. The probabilistic application domain model

The proposed Application Domain is modelled along three different orthogonal adaptivity dimensions: *User's behaviour, Technology* and *External environment,* represented by the *User, Technological* and *External Variables*. The Application Domain Model extends the Adaptive Data Model described in [11] and comprises the following abstract levels:

- *Information Fragments* (or *Atomic Concepts*) like texts, sounds, images, videos, etc. at the lowest level. They can be stored in different data sources and are described by a neutral XML meta-description, produced using *Wrapper* software components.
- *Presentation Descriptions* (*PD*) composed by XML documents, that describe how fragments are to be selected and composed on the basis of different parameters, such as user's profile (computed on the basis of User Variables), Technological and External variables [9]; thus, the system supports a mechanism called *Fragment Variants*. The links in the PDs, also differentiated on the basis of the parameters (to support the *map adaptation*), are annotated by a weight, that represents their importance with respect to each other (*link annotation*). At run time the *Presentation Units* (*PU* or *Pages*) are obtained in a target language (XML, HTML, WML etc) instantiating Presentation Descriptions.
- *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.
- *Application Domain*. Finally, an Application Domain is composed by a set of Elementary Abstract Concepts organised in a digraph; arcs represent relationships between EACs.

Hence, the overall Application Domain with M different profiles (reading keys), can be viewed as a set N of XML documents where the generic document $i \in N$ contains, for each profile k, a set of annotated outgoing links (i, j, k) where j is the destination node. It is mapped in a weighted digraph G where each node corresponds to a XML document and each arc to an outgoing link. We refer to the digraph G 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*.

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:

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

- The mean 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 mean 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.

The intrinsic properties of the structure are used for the construction of a discrete probability density function $s(k)$ which measures the intrinsic relevance of the profiles (see [7]).

2.2. The probabilistic user model

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

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 time spent on recent nodes, $t(R_1), \ldots, t(R_{r-1})$.

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; it is the product of the probabilities associated to the arcs belonging to the R path;
- $\tilde{P}^k_{R_1,R_r}$, the reachability of the next node R_r starting from the first node R_1, through arcs belonging to the profile k; it is the maximum joint probability of all possible paths between R_1 and R_r;
- $D_1[k]$, 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. The page reading times should be accurate; an interesting approach for an accurate computation is proposed in [17].

Those values are used for the construction of a discrete PDF $d(k)$ which measures the "dynamic" relevance of the profiles (see [7]).

The algorithm to compute the new discrete PDF $A'(k)$ on the basis of the user's actions takes as input the User Behaviour Variables (the Technology and External variables are not yet considered), the current $A(k)$ and the initial $A_0(k)$, calculated on the basis of a questionnaire. It computes the new PDF $d(k)$ and then $A'(k)$ is

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

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

The new $A'(k)$ is computed as a weighted mean of terms related to the initial user choices, the current $A(k)$, the browsing history of the single user and the "structural" properties of the hypermedia.

3. System architecture

A system supporting the probabilistic Application Domain and User Models has been designed. The system has a three-tier architecture (Figure 1) comprising the *User,* the *Application* and the *Data Layers.* The *User layer* corresponds to the browser and only receives final pages to be presented.

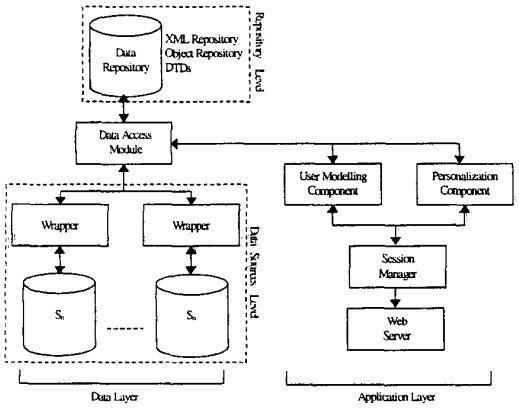

Figure 1. System architecture

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

The main goal of the *Data Layer* is to store persistent data and to offer efficient access primitives. It comprises the *Data Sources Level*, the *Repository Level* and a *Data Access Module*. The Data Sources Level is an abstraction of the different kinds of data sources used to implement the hypermedia.

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* Data Model [12]), persistent objects into an *Object Repository*, and 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.

We are currently implementing a prototype of the proposed system. 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 [13, 16], Java Servlet [15], Enterprise Java Beans, XML [18, 20], and those allowing client-side elaboration (e.g. Java Applet).

In particular, the Author Module has been designed to efficiently support the author of the hypermedia in both the phases of structure definition and content composition.

Furthermore, since it is fundamental to validate the mechanisms which drive the profile assignment decision carried out by the User Modelling Component, a tool for the simulation of the UMC is being implemented. It will allow the author to test the response of the UMC w.r.t. the behaviour of different kinds of users, and the soundness of the overall probabilistic structure of the hypermedia. In this Section we will examine in detail the components of the Author Module and show structure and features of the simulation tool.

3.1. The author module

The Author Module (Figure 2) allows the author of the hypermedia to browse the data sources and metadata associated to them, and to define in a natural way the structure of the hypermedia and the layout of the Presentations w.r.t. the supported adaptivity dimensions. Furthermore, the Author Module allows to validate (with respect to syntactic and semantic correctness) the XML documents representing the Presentation Descriptions and the persistent objects representing the overall hypermedia structure.

The main components of the Author Module are:

- The *Hypermedia Modeller*, which allows to design the adaptive hypermedia as a digraph of EACs. Moreover, it allows to design EACs as weighted digraphs of XML descriptions. In particular, it allows to define the probabilities of the arcs and offers a set of utilities such as computation of the resulting PDF $s(k)$, search of shortest (maximum weight) 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 Repository Level (Object Repository). The persistent objects also contain additional data, e.g. the shortest path for each pair of nodes and the PDF $s(k)$ (see Section 2); the utility emphasises the existence of some high-probability cycles and asks for

confirmation. The use of a persistent representation allows to reuse (part of) the hypermedia, e.g. some EACs, for different web systems.

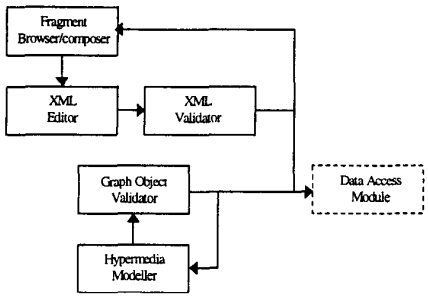

Figure 2. Author Module

- The *Fragments Browser/Composer*, which allows the browsing of information fragments provided by the heterogeneous data sources and XML Metadata 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 editor also allows a "preview" of the Presentation Units.
- The *XML Validator*, which performs a validating parsing of XML documents with respect to the DTDs and stores them into the repository.

3.2. A tool for the simulation of the system

The Simulation Tool (Figure 3) 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:

- analyses the $s(k)$ calculated from the structure of the hypermedia;
- defines a set of *User Classes* that describe the behaviour of typical users;
- analyses the response of the UMC w.r.t. the User Classes.

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 the *User Generator*), communicating them to the User Modelling Component, and storing the history of such requests, together with the response of the UMC.

For each user, the AHS Simulator stores a sequence of objects describing each choice made by the user and responses of the UMC; each object contains:

- the time of the choice;
- the chosen link;
- the current profile;
- the current PDFs $t(k)$, $r(k)$, $c(k)$, $d(k)$, $A(k)$.

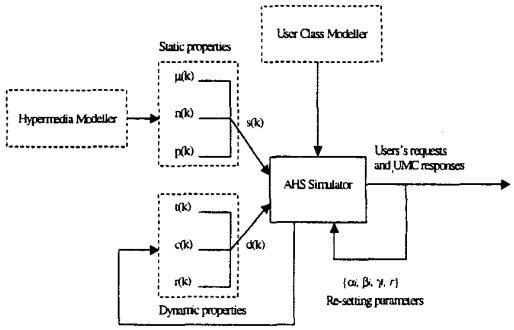

Figure 3. Simulation tool

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. A User Mask can express the following characteristics:

- the user can be *Unchanging* or *Changing*: his/her behaviour can refer constantly to one profile or can change at each choice, with a random distribution.
- the choice of the arc can be *Random*, *Minimum*, *Mean*, *Maximum* w.r.t. the probabilities of the arcs in the user's current profile (i.e. w.r.t. the visibility of the links in the page). If the choice of the arc is random, the distribution can respectively be *Uniform w.r.t. the number of the arc* (e.g. 0.25 from 1 to 4 if there are 4 arcs), *Uniform w.r.t. the probability of the arc* (e.g. 0.1 from 0 to 1 if there are 10 arcs, and the chosen arc is the one with the nearest probability value), *Gaussian w.r.t. the probabilities of the arc*.
- the time spent on the nodes can be *Constant*, *Decreasing*, *Increasing*, *Random*. If the time spent on the nodes is random, the distribution used can be *Uniform*, *Exponential*, *Erlangian*, *Hyperexponential*, *Gaussian*; the values characterising each of these distributions can also be defined.

The author can also specify the number of users to simulate for each class. It should be noted that classes of "deterministic" users (i.e. unchanging users with deterministic choices of arcs and times on the nodes) must contain exactly one user, since their kind of interaction does not add any extra information over repeated simulations.

At the end of the simulation (i.e. when the temporal duration is completed), the AHS Simulator shows, also graphically, the history of the PDFs (of their mean values in the case of non-deterministic users); it allows to analyse their trend w.r.t. profiles (e.g. how an high-probability link for a profile can increase the corresponding probability in a single "jump") 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 (in this case the simulation must be re-executed);

- change the length of the sliding temporal window, r (the simulator re-evaluates also the PDFs $A(k)$, $d(k)$, $t(k)$, $r(k)$, $c(k)$);
- change the values of the parameters α_i, β_i and γ_i (in this case, only the values of $A(k)$ and $d(k)$ are re-computed).

4. Conclusions and future work

We presented the design and development of a Probabilistic Adaptive Hypermedia System. The system uses a layered data model to describe the Application Domain and a probabilistic approach to adapt the hypermedia contents and links to the user's behaviour.

The probabilistic User Model combines 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.

The AHS Simulator is currently used to test and verify the UMC behaviour with respect to predefined classes of hypothetical users to allow the restructuring of the hypermedia (graph structure and weights). We plan to use the core of the simulator into the run-time support of the system to dynamically change the hypermedia structure and the supported user classes (profiles), on the basis of the real traffic on the system.

Moreover, we will enhance the support of the *technology* and *external environment* adaptivity dimensions.

Acknowledgements

This work has been supported by EC FESR Project "Sviluppo di tecnologie digitali di grafica avanzata per applicazioni industriali e commerciali".

References

[1] S. Abiteboul, B. Amann, S. Cluet, A. Eyal, L. Mignet, T. Milo, "Active views for electronic commerce", *VLDB* 99.

[2] 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.

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

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

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

[6] 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.

[7] M. Cannataro, A. Cuzzocrea, A. Pugliese, "A probabilistic approach to model adaptive hypermedia systems", in *Proceedings of the International Workshop for Web Dynamics*, 2001.

[8] 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.

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

[10] 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.

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

[12] S. Flesca, S. Greco, E. Zumpano, "Modelling and querying XML data", *IEEE IDEAS 2000 Conference*, Yokohama, Japan. IEEE Computer Society Press, 2000.

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

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

[15] MageLang Institute, *Fundamentals of Java Servlets*, Short course, 1999.

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

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

[18] B. Wait, "Using XML in Oracle database applications", Technical Information, Oracle Corporation, 1999.

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

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